(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 May 2001 (17.05.2001)

PCT

(10) International Publication Number
WO 01/35200 A1

(51) International Patent Classification⁷: G06F 1/32, 1/20

(21) International Application Number: PCT/US00/17460

(22) International Filing Date: 23 June 2000 (23.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/164,409 9 November 1999 (09.11.1999) US
09/477,141 3 January 2000 (03.01.2000) US

(71) Applicant: ADVANCED MICRO DEVICES, INC. [US/US]; One AMD Place, Mail Stop 68, Sunnyvale, CA 94088-3453 (US).

(72) Inventors: BAUM, Gary; 189 Ehrlich Road, Austin, TX 78746 (US). ODIORNE, Kyle; 10806 Redmond Road, Austin, TX 78739 (US). MITCHELL, Charles, Weldon; 6501 Skinner Cove, Austin, TX 78759 (US). QURESHI, Qadeer, Ahmad; 16708 Tomcat Drive, Round Rock, TX 78681 (US). CALDWELL, Dervinn, Deyual; 3837 Burton Common, Fremont, CA 94536 (US).

(74) Agent: RILEY, Louis, A.; Advanced Micro Devices, Inc., 5204 East Ben White Boulevard, M/S 562, Austin, TX 78741 (US).

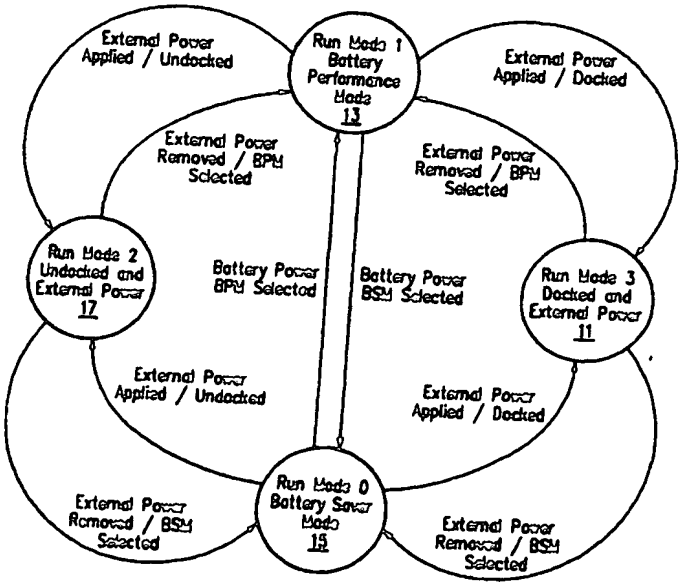
(81) Designated States *(national)*: JP, KR.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DYNAMICALLY ADJUSTING A PROCESSOR'S OPERATIONAL PARAMETERS ACCORDING TO ITS ENVIRONMENT



(57) Abstract: When a computer system detects a change in one of a plurality of operating characteristics, the system stops core clocks running on the processor. Updated frequency control information is provided to clock control logic in response to the detected change and updated voltage control information is supplied to a voltage control circuit in response to the change. Once the updated information has been provided, the system restarts the clocks to operate the processor at a second clock frequency corresponding to the updated frequency control information and at a second voltage corresponding to the updated voltage control information.

DYNAMICALLY ADJUSTING A PROCESSOR'S OPERATIONAL PARAMETERS ACCORDING TO ITS ENVIRONMENT

Technical Field

This invention relates to portable computers and performance and thermal issues associated therewith.

Background Art

A conventional notebook computer has power and thermal constraints that cause it to operate at performance levels below an equivalent desktop computer. When using a battery as a power source, a conventional notebook computer often employs techniques to conserve battery life, which can reduce performance levels. In addition, the conventional notebook computer has a small, densely packed system construction that limits its ability to safely dissipate the heat generated by computer operation. Therefore conventional notebook computers generally use less power than their desktop counterparts, which adversely affects performance.

Many power saving techniques have been introduced to try and mitigate the limitations caused by thermal and battery power constraints. The frequency of operation (clock frequency) of the processor and its operating voltage determines its power consumption. Since power consumption and therefore heat generation are roughly proportional to the processor's frequency of operation, scaling down the processor's frequency below desktop performance levels has been a common method of staying within notebook computer power limitations.

A common power management technique called "throttling" prevents the processor from over heating by temporarily stopping processor operations by stopping processor clocks. Throttling is an industry standard method of reducing the effective frequency of processor operation and correspondingly reducing processor power consumption by using a clock control signal (e.g. the processor's STPCLK# input) to modulate the duty cycle of processor operation. A temperature sensor placed on or near the processor initiates throttling when needed. Throttling continuously stops and starts processor operation according to a predefined duty cycle with a period of a few milliseconds. The reduction in the effective speed of the processor reduces power dissipation and thus the processor's temperature.

Applications like word processors typically leave the processor idle much of the time. As a result, the typical processor power consumption when running word processing applications can be as much as 30-50% below the maximum. That idle time can be exploited by the computer system to achieve additional power savings by putting the processor to sleep temporarily.

For example, in a word processing application, a processor will do a brief burst of work after each letter is typed, then its operation is stopped until the next keystroke. Additionally, peripheral devices may be turned off to obtain move power savings. For example, the notebook's hard drive may be suspended after a certain period of inactivity until it is needed again. If the system detects another period of inactivity, e.g., a few minutes, the display may be turned off. Such techniques are useful in conserving battery power and in the

case of the processor, reducing the amount of heat needed to be dissipated. It is also common practice to use a cooling fan to increase the amount of heat removed from the system, lower processor temperature and prevent damage to the system.

A typical notebook computer, with power management active and operating from its battery, consumes about 15 to 20 Watts. The processor portion of the power budget is typically 8 – 12 watts. The remaining power budget goes to the display, hard drive, memory subsystem, graphics controller and other peripherals. With a 40 to 50 watt-hour battery pack, the notebook will run for 2.5 to 3.5 hours. In contrast, without those power and thermal constraints of the notebook processor, typical desktop processors consume 20-30 Watts.

However, when a notebook computer is plugged into AC-line power, ample system power is available and auxiliary cooling capacity can be made available to remove more heat from the processor. That can allow the processor to operate at a performance level approaching that of a desktop processor. In such an environment, with all power management features disabled, the entire notebook system may stay fully operational. Total notebook power consumption and dissipation can increase to about 20 to 30 Watts.

Appropriately monitoring and controlling the processor's operating parameters is important to optimizing the notebook's performance and battery life. Power management in older personal computer systems was typically implemented using micro-controllers and/or proprietary use of the system management interrupt (SMI). Current x86 based computer systems utilize an industry supported power management approach described in the Advanced Configuration and Power Interface Specification (ACPI), Revision 1.0a, by Intel, Microsoft and Toshiba dated November 19, 1998. The ACPI is an operating system (OS) controlled power management scheme that uses features built into the Windows 9x and Windows NT or other compatible operating systems. It defines a standard interrupt (System Control Interrupt or SCI) that handles all ACPI events. System control interrupts are generated by devices to inform the OS about system events.

As part of that power management approach, ACPI specifies sleep and suspend states. Sleep states temporarily halt processor operation and operation can be restored in a few milliseconds. A notebook enters the sleep state when internal activity monitors indicate no processing is taking place. When a keystroke is entered, a mouse moves or data is received via a modem, the processor wakes up.

Suspend states shut down more of the subsystems (e.g. display or hard drive) and can take a few seconds for operation to be restored. Suspend states may copy the present context of the system (sufficient for the computer to resume processing the application(s) presently opened) into memory (suspend to RAM) or to the hard drive (suspend to disk) and power down peripherals.

ACPI defines standard power states of a system as well as the power states of individual components. In addition, it defines standard ways of putting system and devices into different power modes, and has features to allow reporting events, monitoring and controlling temperature in a system, and monitoring battery. ACPI power management includes the many system power states for notebook PC operation. The Gx states indicate the overall operational status of the system. Cx processor states, Dx device states, and Sx sleep states

define the status of the subsystems and the sleep states, such as suspend-to-RAM and suspend-to-disk. The four global Gx states are shown in Fig. 1A.

When the computer is operating in the G0 state, the CPU can have the four computing states shown in Fig. 1B. Note that in a given platform, it may not be necessary to support all the CPU states. For example, C1 and C2 for some systems may offer similar power dissipation and latencies for restoring operation. Therefore, the designer may choose to implement only one of these states. Also, different systems may have different implementations of a particular CPU state. Unlike the sleep states, in which various parts of the computer system may be powered down, all systems remain powered up in the computing states.

When the computer is in the G1 state (sleeping), the system will be in one of the four sleep states shown in Fig. 1C. Like the Cx states, some sleep states may not exhibit significantly different behavior for a given design. Therefore, it may be reasonable to not implement one or more of the states in a particular system.

As can be seen, the ACPI environment provides a number of mechanisms to deal with thermal and power issues. However, the desire for notebook performance to approach that of desktop computers requires the processor to run faster and dissipate more heat. However, the notebook still must run in the mobile environment limited by power and thermal constraints. Therefore, it would be desirable if the notebook could adapt readily to its environment in order to provide the appropriate level of performance given the operating environment. While ACPI and current power management techniques provide some level of monitoring and control based on a notebook computer's operating environment, there is a need to provide improved power management techniques that more effectively responds to the environment in which the notebook computer is being used. Further, it would be desirable to dynamically adjust to the demands of the various environments without significantly impacting the user.

DISCLOSURE OF INVENTION

Accordingly, a notebook or similar computing device monitors system environment such as availability of external power sources (AC adapter, auto adapter or other external power source), attachment and/or activation of auxiliary cooling devices, and a profile, which may be user definable, for choosing performance criteria during battery operation. When changes take place for any of these factors, system level software assigns appropriate operating parameters or "run states" for the processor of the computing device.

Each run state has different limitations on available power and power dissipation and the situation can change while the notebook is in use. Ideally, in each run state the processor takes full advantage of the available power and the power dissipation ceiling. In order to change the run state, the system changes the processor's core clock frequency and core voltage. Since processor frequency determines a minimum required voltage for operation, the voltage and frequency of operation for the processor core are changed at the same time. As core frequency is changed, the core voltage is also changed.

In one embodiment a method is provided for controlling the power consumption of an integrated circuit in an electronic system. The method includes operating the integrated circuit at a first voltage and at a first frequency. When the system detects a change in at least one of a plurality of operating characteristics in the electronic system, in response to detecting the change, the system stops clocks running on at least a substantial portion of the integrated circuit. Updated frequency control information is provided to clock control logic in response to the detected change and updated voltage control information is supplied to a voltage control circuit in response to the change. Once the updated information has been provided, the system restarts the clocks to operate the integrated circuit at a second clock frequency corresponding to the updated frequency control information and at a second voltage corresponding to the updated voltage control information.

Another embodiment provides a computer system that includes an integrated circuit having a first logic portion coupled to receive a first clock and a first voltage. A programmable voltage regulator circuit supplies a variable voltage level for the first voltage according to voltage control signals provided to the voltage regulator circuit. A clock control circuit generates the first clock at a frequency determined according to frequency control signals. A control circuit receives an indication of changes in a plurality of operating characteristics in the computer system. The control circuit responds to a change in one of the operating characteristics by providing updated voltage control signals and frequency control signals indicating a new voltage value for the first voltage and a new frequency for the first clock. The new voltage value and the new frequency correspond to the detected change in the operating characteristic.

BRIEF DESCRIPTION OF DRAWINGS

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

Fig. 1A is a table showing the four global Gx states.

Fig. 1B is a table showing the four computing states.

Fig. 1C is a table showing various sleep states.

Fig. 2 illustrates a state machine implementing various run modes allowing the processor to dynamically adjust to its environment.

Fig. 3A summarizes the various run modes illustrated in Fig. 2.

Fig. 3B provides exemplary performance parameters for various run modes.

Fig. 4 provides a graph illustrating relationships between voltage, frequency and power.

Fig. 5 illustrates a high level diagram of a computer system incorporating one embodiment of the present invention.

Fig. 6 illustrates one implementation of a clock control circuit in the CPU.

Fig. 7 illustrates generally the operation to control voltage and frequency according to one embodiment of the present invention.

Fig. 8 illustrates the use of power on suspend CPU context lost (POSCCL) in one embodiment of the invention,

Fig. 9A shows timing charts illustrating a POSCCL suspend and the corresponding resume operation.

Fig. 9B is a table describing the signals shown in Fig. 9A.

Fig. 10 illustrates use of a programmable logic device to effect run mode changes.

Fig. 11 illustrates an implementation of a logic device from Fig. 10.

Fig. 12 is a timing diagram illustrating the operation of logic device of Fig. 11.

Fig. 13 shows a run mode control register utilized in one south bridge implementation to effect run mode changes.

Fig. 14 shows a high level block diagram of a south bridge implementation to provide both jumper inputs and register inputs for voltage and frequency control.

Fig. 15 illustrates a flow chart implementing run mode changes in a south bridge integrated circuit.

Fig. 16 illustrates an exemplary docking station which may be used with a notebook computer incorporating the various run modes described herein.

The use of the same reference symbols in different drawings indicates similar or identical items.

MODE(S) FOR CARRYING OUT THE INVENTION

A notebook computer or other portable computing device according to one embodiment of the present invention, dynamically adapts the operation of the notebook computer and its processor to changes in its environment to provide improved performance and battery life. To determine those changes, the notebook computer monitors such things as application or removal of external power sources (AC adapter, auto adapter or other external power source), changes in thermal environment (attachment of auxiliary cooling devices embedded in AC adapters, port replicators, docking stations or other attachable devices with cooling capabilities or cooling capability within a notebook that can be used because of the availability of external power), and changes in a user definable profile for battery operation (e.g. maximizing performance or battery life).

When changes are detected for any of those or other parameters, the notebook computer adapts to the change by entering an appropriate run mode which sets the processor's frequency of operation, operating voltage, power consumption and power dissipation capabilities. That may be accomplished by generating an interrupt when parameter changes are detected, that causes system level software to be executed that assigns appropriate operating parameters for the various run modes of the computing device. Preferably, the notebook computer makes appropriate changes dynamically without requiring the user to exit application programs or system software.

The various run modes reflect the different environments in which a notebook has to operate. For example, in some environments, such as when running on battery power, battery life may be more important than performance. However, while playing a video clip, performance is probably more important. Plugged into an AC-adapter or auto-adapter, battery life is not an issue. Ideally, in each run mode the processor takes full advantage of the available power and the power dissipation ceiling.

CPU thermal and power management are improved by changing CPU voltage in addition to changing the clock frequency. Each run mode matches processor frequency and voltage of operation parameters to dynamic changes in the performance requirements, power consumption limitations and power dissipation limitations to provide improved performance and battery life to the user. Since processor frequency determines a minimum required voltage for operation, the voltage and frequency of operation for the processor core are changed at the same time. Thus, reducing voltage along with frequency is a highly effective way of reducing the power consumption of the system's processor when the environment is power or thermally constrained or when power conservation is desired. Battery life is enhanced by allowing the voltage provided to the CPU to be the least possible to assure proper operation at the target frequency of operation. In effect, this enables the lowest possible CPU power consumption at a given frequency of operation. The system is now able to optimize power and frequency within specified limits. In addition, thermal management is now optimized for a given frequency of CPU operation.

Changes in the processor's core clock frequency have an approximately linear affect on the power dissipated by the processor. Thus, a 20% reduction in clock frequency reduces the power dissipated by the processor by 20%. The range of change is significant since a ratio of lowest frequency to highest frequency is usually greater than 2:1. Consequently, the processor's power may be changed by similar ratio. Changes in the processor's core voltage have an approximately square law effect. That is potential power savings is proportional to the square of the percentage of voltage reduction. Although the range of change of voltage is generally less than 50%, the square law effect results in significant changes in the processor's power if the core voltage of the processor is reduced. Note that high performance processors typically receive multiple voltages including a voltage for the I/O region and a voltage for the core logic region, the core logic voltage typically being less than voltage required in the I/O region because the I/O region requires sufficiently high voltage to drive signals off chip.

Referring to Fig. 2, a state machine is shown that implements the various run modes allowing the processor to dynamically adjust to its environment. Because it is desirable for a notebook to operate with the

performance of a desktop computer, run mode 3 (11) provides maximum system performance (clock frequency and heat dissipation) when, e.g., docked in a docking station providing auxiliary power and auxiliary cooling. That may require that the docking station incorporate a sophisticated cooling system to force air through the processor heat sink or to otherwise conduct heat out of and away from the processor with, e.g., a heat pipe or heat plate. The heat pipe may also be used to conduct heat into the docking station where a heat sink and fans are used to dissipate the heat.

If the system becomes undocked, the system can enter either run mode 1 (13) or run mode 0 (15). When operating from the battery (run mode 0 or 1), the user can choose between power conservation and performance. Run mode 1 is a performance mode that maintains processor speed that may be as high as run mode 2 and require active cooling. That high level of performance reduces battery life as a result of increased power dissipation by the processor and by the need to run a cooling device such as a fan to aid in heat dispersion. In the performance/battery operation mode, the performance is limited by the limits of active cooling.

Alternatively in run mode 0, battery saver mode, battery life is emphasized over performance. In the maximum battery life mode, run mode 0, the limits of passive cooling provide a performance ceiling. That ceiling may be higher than actual performance due to the desire to extend battery life by reducing power consumption by the processor even below the performance ceiling. The ability to operate without active cooling in the Battery Saver Mode (Run Mode 0) is dependent on low power dissipation in the Stop Clock/Grant state in which processor core clocks are stopped. Otherwise power will have to be expended for active cooling.

In addition, at least one mode of operation may be provided that is between the two extremes of run mode 1 and run mode 0. That "between the two extremes" mode provides active cooling but with a lower performance target that results in active cooling needing to switch on less frequently. That operational mode benefits from lower processor power consumption and less frequent consumption of power by the cooling fan. Additional battery modes may be provided which have even more granularity between the performance emphasized in run mode 1 and the battery life emphasized in run mode 0. In one implementation, the user may specify through a control panel applet the various battery operation modes in a manner similar to a user selecting the time delay before the display or hard drive sleeps.

Run mode 2 (17) provides external power (e.g., from an AC adapter) while the notebook computer is undocked. Run mode 2 provides for maximum performance limited by thermal considerations. The lack of auxiliary cooling may limit run mode 2 performance below run mode 1. However, if the notebook has an active cooling device, it can be utilized continually in run mode 2 without concerns about power consumption. That allows the CPU to operate at a higher frequency than in run mode 1.

Each run mode is intended to provide the maximum performance within the constraints of the available cooling mechanism and battery life requirements. Fig. 3A summarizes the various run modes illustrated in Fig. 2 along with total dissipated power (TDP). It is possible for various environments to prompt

or warn the user about operating. For example, a DVD movie playback program can check the run mode when launched. ACPI maintains tables indicating the performance level. If operating in run mode 0 (battery saving mode), a warning message can be generated that the full frame rate of playback may not be possible until the user selects one of the other operating modes.

The computer system should provide a small latency between run modes. The user may be able to tolerate a latency of up to, e.g., 1 second but preferably the latency should not be noticed by the user.

Fig. 3B provides a table of exemplary performance parameters for various run modes. For example in run mode 0, CPU voltage is 1.6 volts and the CPU frequency is 200 MHz. In contrast, run mode 3 provides 400 MHz operation at 2.2 volts.

Referring to Fig. 4, the graph shows a comparison of power reduction for a notebook that reduces its average frequency of operation through "throttling" which was described previously. The left vertical axis is in volts. The right vertical axis is in Watts. Line 41 illustrates the voltage required as a function of frequency for a typical notebook processor. The middle line 43 shows power as a function of frequency and illustrates the power savings available from reducing frequency. As can be seen the power savings is generally linear. Note that the power savings from reducing frequency is equivalent to power savings provided from throttling. Line 45 shows power as a function of both voltage and frequency and illustrates the power savings available from reducing both voltage and frequency. Note that throttling may be used in combination with the reduction of both voltage and frequency to further reduce the effective speed or power dissipation of the processor. In a typical notebook system, the additional power savings at 200 MHz is equivalent to at least 45 minutes of battery life.

Run mode changes are controlled by state machine logic that is triggered by software but once triggered the state machine can perform its operations while the processor is sleeping. The run mode logic can be built into the power management features of a south bridge integrated circuit or implemented with a separate logic device that augments standard south bridge power management or in any other location suitable in the computer system. The software required to make run mode changes can be triggered by SMI or SCI features built into standard south bridges. The software required can leverage existing routines for placing the processor in a sleep or suspend mode and then resuming operations.

In one embodiment as described further herein, a processor changes its internal bus-multiplier state and maintains or recovers the state of its internal registers through chipset control. That feature allows for multiple modes of frequency operation without powering-off the system or manually reconfiguring the bus frequency (BF) pins as described further herein.

Because sleep and suspend states require the processor operation to be stopped, it is impossible for system software to control all sleep, suspend and recover operations. To overcome this problem, state machines are provided in the input/output integrated circuit (known as the south bridge) to control the final stages of sleep and suspend operations and the resume operation. Many notebook computers use state machines within the south bridge integrated circuit to provide common power management features . One

such integrated circuit is the 82371AB PCI-TO-ISA/IDE XCELERATOR (PIIX4) available from Intel Corp. The power management features contained therein reduce power consumption to extend battery life and control heat generation and dissipation to safely operate the processor. While some use a separate microcontroller for the task, most notebook PCs rely on the south bridge to provide the hardware for thermal and power management. The south bridge is one chip of a chipset that also includes a north bridge. The north bridge provides a memory controller function as well as a bridge function between the Peripheral Component Interconnect (PCI) bus and the host bus connected to the processor. The south bridge also interfaces to the PCI bus (which functions as a major input/output bus in the computer system) and provides a variety of functions including providing an interface with legacy devices on the ISA bus (or integrated into the south bridge), providing an interface to various other input/output buses and/or functions (e.g. Universal Serial Bus (USB)) and also providing various power management related functions. South bridge chips from various manufacturers have typically utilized the registers, timers and state machine definitions used in the Intel PIIX4 south bridge. PIIX4 compatibility in current south bridge chips can be extended to support mobile operational modes described herein.

In an exemplary embodiment illustrated in Fig. 5, voltage regulator 501 supplies core voltage 502 (commonly referred to in the x86 processor environment as V_{CC2}) to processor (CPU) 503. In the embodiment illustrated, south bridge 505 controls the voltage level that is supplied to CPU 503 by supplying voltage control signals VID[4:0] to voltage regulator 501. In order to control processor frequency, in one typical implementation such as that used for the AMD-K6® processor, three bus frequency input pins (BF[2:0]) are used to determine the internal operating frequency of the processor. South bridge 505 controls the operating frequency of CPU 503 by supplying the CPU with BF (bus frequency) signals BF[2:0]. The bus clock signal 506, which is supplied to CPU 503 from clock generator 507, is multiplied internally by the CPU by a ratio determined by the value of the three bus frequency pins. The multiplication factor ranges, in one implementation from 2.5 times the bus clock to 6.0 times the bus clock. Other multiplication factors are possible according to the specific system implementation.

Fig. 6 illustrates one implementation of the clock control circuit in the CPU. Frequency divider circuit 61 receives the BF pins, which are sampled during assertion of a processor reset signal (CPURST). The sampled values are applied to a phase locked loop (PLL) clock multiplier/synthesizer circuit for a long enough period of time for the PLL to stabilize. The values of the BF pins are latched into frequency divider 61 on the falling edge of the CPURST signal. The reset pulse is sufficiently long to ensure that the clock multiplier circuitry is stable. The bus clock 63 is provided to the phase (frequency) detector 64 which provides a control voltage 65 to voltage controlled oscillator (VCO) 67. The VCO supplies the core logic of the CPU with a clock having a frequency determined by the bus clock frequency multiplied by a value determined from the BF pins. Gating logic 68 may be used to gate off the CPU core clocks when the appropriate gating signal 69 is asserted to stop core clocks.

Additional BF pins may be useful to provide greater range in clock multiplier values to ensure that battery life mode (run mode 0) is adequately supported, i.e., the processor can run slow enough, as faster and faster processors are provided.

Referring again to Fig. 5, south bridge 505 supplies enable signal 509 to clock generator 507 to shut off the clock supplied to CPU 505 entirely to minimize the power consumed by the CPU. Thus, south bridge 505 programs the voltage regulator, controls the clock generator, manages the duty cycle for throttling of the processor via the STPCLK# signal 510 and controls the clock multiplier by controlling the BF-pins. In addition, the south bridge 505 orchestrates the special protocol required to change the clock multiplier in the processor and change the core voltage in order to manage the transitions between the run modes. In embodiments where direct operating system (OS) support for changing run modes is not provided, the transition between run modes is preferably transparent to both the operating system and the user to the extent possible.

One way in which transparency can be achieved is for a system management interrupt (SMI) to be used when a change in docking or AC/battery status is detected. That interrupt is unused by the ACPI scheme and is transparent to the operating system. When a change is detected, a short latency of up to about 1 second for the change is acceptable but a shorter latency is desirable. Note that operating system support may be provided in some embodiments and thus transparency ceases to be advantageous.

As shown in Fig. 5, jumpers 511 provide a default state for the clock multiplier and voltage regulator on bootup. The south bridge receives the jumper signals at eight jumper inputs pins IBF[0:2] and IV[0:4]. The number of jumpers and jumper inputs may vary according to the particular design. The settings of the jumpers 511 with resistors 513 provide the default values for both the voltage regulator control pins Voltage ID (VID) pins and the processor clock multiplier pins (BF pins). Those default values are provided to south bridge 505. On power on, the south bridge provides the default values of the VID pins and the BF pins to the voltage regulator and the CPU, respectively. However, to transition between run modes described herein, south bridge 505 according to one embodiment of the present invention, selects internal registers as the source for the output signals provided to the voltage regulator and CPU rather than the jumper inputs as described further herein.

In the embodiment illustrated in Fig. 5, three output bits are used to control the clock frequency and five output bits may be used for control of the CPU core voltage regulator. However, other numbers of bits may be used according the particular clock frequency approach and voltage regulator being used. In addition, particular bits need not be dedicated as voltage or frequency control bits. Thus, if the south bridge provides for ten inputs for the jumpers and ten outputs for clock and frequency control, some applications may require only three frequency control pins and 5 voltage control pins while others may require four of each or five of each. Further, the input bits and outputs are not dedicated as a frequency or voltage control bit. Therefore, if the voltage regulator can utilize seven control bits, seven control bits may be utilized for voltage control and three bits for frequency control. That advantageously provides flexibility in implementation without having to change the south bridge. In some implementations, the processor may supply default static VID signals such as those shown at 514 rather than relying on jumper settings.

In order to change both the voltage and the frequency of the processor, processor operations need to be stopped, i.e., processor clocks need to be stopped, at least those clocks supplied to storage elements such as

registers and latches or other circuit nodes in time sensitive paths, since otherwise unpredictable behavior may result. In current X86 architectures that is achieved by first asserting the STPCLK# signal to cause the CPU to stop its internal clock distribution. On receipt of STPCLK#, the CPU completes the currently active instruction and asserts a "Stop Grant" indication. Once the "Stop Grant" is received, clocks may be stopped at clock generator 507 using enable signal 509. Conveniently, the framework of ACPI provides a number of suspend and sleep operations which are supported in the south bridge and which may be modified to implement suspend processor operation in the context required here.

The flow chart in Fig. 7 illustrates generally the operation to control voltage and frequency according to one embodiment of the present invention. Assume that the processor is operating in a normal operational mode in 70 with the voltage regulator 501 receiving appropriate voltage control signals VID[0:4] and frequency control circuitry receiving appropriate frequency control signals (e.g. the BF signals). The computer system detects a change in operating characteristics such as power source, thermal environment or user selected operating parameters in 71. In response to detecting that change, the system stops processor operation in 72 by stopping its clocks and determines a new frequency and corresponding voltage settings appropriate for the new run mode. That information may be saved in registers in the south bridge or in other suitable locations in the computer system. Updated voltage and frequency control signals are supplied to the appropriate voltage and frequency control circuit in 73 and 74 and then the processor resumes operation in 75.

The resume operation is triggered from one of the last actions of the suspend operation. One of the steps in the resume operation contemplated in association with Fig. 7, is for the resume operation to issue a frequency control signal update indication (e.g., a reset (CPURST)) to the processor. That frequency control update or valid signal indicates to the processor that there is valid data on the BF pins that should be used for generation of core clocks. If a CPU reset is used for that purpose, that reset is applied only to the processor and not to those portions of the computer system not requiring a reset.

In one implementation, the processor senses the values of the frequency control signals (on the BF pins) when a reset signal is asserted and latches in those values on the falling edge of reset (see Fig. 6). Alternatively, a signal other than reset may be asserted to indicate to the processor that new frequency control signal values are present. If the reset signal is used to indicate that new frequency control (BF) signals are available, the processor loses context on assertion of reset, e.g., values in the processor registers may be lost. Thus, processor context should be saved prior to issuing a reset or an application will not be able to resume where it left off. Once the reset is completed, the processor context can be restored from wherever it has been saved with the processor operating at the new frequency and voltage settings. It is desirable to minimize the latency of the resume operation and therefore desirable to restore the processor context as fast as possible.

Sleep and suspend states require the processor operation to be stopped. Therefore, it is impossible for system software to directly control some management operations. To overcome this problem, state machines in the south bridge take control of the system to suspend processor operation and suspend other system devices. Once the system is in a sleep or suspend state, the south bridge monitors several possible events for waking the system. When an events occurs, another state machine sequences the system to resume operation.

The same hardware and software used to provide the sleep and suspend states can be utilized in performing the switch between run modes.

As previously described, the processor's STPCLK# (the # sign indicates the signal is active low) input is often used to temporarily suspend operation and conserve power. Use of the STPCLK# signal allows the processor to be put in a Stop Grant state. In that state, the core clocks are stopped although some minimum logic including clock multiplier logic still operates. To start the sequence used by current south bridge chips to place a Socket-7 processor in the Stop Grant state, a control register (LVL2) in the south bridge is read. That results in STPCLK# being asserted which notifies the processor it should stop processor clocks. The south bridge waits for the processor to complete a current operation and issue the Stop Grant indication thereby indicating that the processor has gated off its core clocks. The south bridge asserts the ZZ pin (which is optional to suspend L2 SRAM). That is enabled by ZZ_EN in CNTB register in the south bridge.

When a wake up event is detected, the south bridge deasserts the ZZ pin (if this option was enabled) and deasserts STPCLK# . Using this control sequence, which can save significant power while waiting for an event like the next keystroke, processor clocks can also be stopped for changes of the core frequency associated with changes in run modes as described herein. A variant of this sequence asserts the SLP# signal to put portions of a compatible processor into a powered down state.

More power can be saved by also stopping the clock being supplied to the processor clock multiplier circuitry to eliminate that portion of the processor still active as described previously. Clocks at clock generator 507 may also be stopped as part of the sequence to change run modes as described herein. To start the sequence, in a PIIX4 compatible south bridge, to place a processor in the deep sleep state with the processor clock off, software reads a control register (LVL3) in the south bridge, which results in STPCLK# being asserted. The south bridge waits for Stop Grant. The ZZ pin is optionally asserted to suspend L2 SRAM. That is enabled by ZZ_EN in CNTB Register. Then SLP# is asserted. SUS_STAT1# is asserted to the North Bridge to place system memory is auto-refresh mode. CPU_STP (enable 509 in Fig. 5) is asserted to disable the clock synthesizer (clock generator 507) output for the CPU bus clock.

When a wake up event is detected, the south bridge first deasserts CPU_STP to enable the CPU bus clock output of the clock synthesizer. Then a timer (known in the industry as the "Fast Burn Timer") in the south bridge counts down allowing time for the CPU PLL to lock. Note that the value used by the timer is loaded from the CLK_LCK register which is set by the BIOS during the system's power-on self-test (POST) sequence. Finally, SUS_STAT1#, SLP#, the ZZ pin (if this option was enabled), and STPCLK# are de-asserted.

The sleep state machine can perform the more complex operations necessary to suspend the system by setting SUS_EN (bit 13) and loading the appropriate value (bits [12:10]) in the south bridge's Power Management Control Register. The values indicate the type of suspend/resume operation desired. Table 1 below details the type of suspend/resume operations available and their associated values in the Power Management Control Register in the south bridge. The resume latencies vary depending on the type of

suspend operation. For example, the suspend to disk resume latency is typically less than 30 seconds, suspend to RAM approximately one second and power on suspend with context maintained approximately 20 ms. Maintaining context obviously reduces resume latency.

Table 1

Bits [12:10]	Suspend Type
0 0 0	Soft Off or Suspend to Disk (STD)
0 0 1	Suspend to RAM (STR)
0 1 0	Powered On Suspend, Context Lost (POSCL)
0 1 1	Powered On Suspend, CPU Context Lost (POSCCL)
1 0 0	Powered On Suspend, Context Maintained (POS)
1 0 1	Working (Clock Throttling May Be Active)
1 1 0	Reserved
1 1 1	Reserved

As previously discussed, asserting reset to notify the processor of a change in BF signal values to effect clock frequency changes associated with changes in run modes, causes the processor context to be lost. One of the suspend operations which restores CPU context is the Powered on Suspend, CPU Context Lost (POSCCL) shown above. That suspend operation may be utilized when a reset is used to latch in the new BF pin values as illustrated in Fig. 8.

Referring to Fig. 8, which illustrates the use of POSCCL in one embodiment of the invention, once the new operational environment is detected, i.e., a new run mode is desired, the new voltage and frequency settings are loaded into appropriate registers in 801. Those registers may be in the south bridge or in additional logic, which embodiment is described further herein, or in any other suitable location in the computer system. Once those registers are loaded, the processor context is saved and the resume operation is set up in 803 by specifying a restart address. Saving processor context may include flushing the internal cache of the processor and saving the state of the processor to DRAM. The set up of the jump entails setting the start-up vector address (real mode) and setting the necessary flag byte so the BIOS will immediately branch to the restore routine after CPU reset instead of rebooting the system. X86 processors always begin execution in the address range F000:FFF0 after reset. In a conventional x86 personal computer system, the BIOS ROM exists at that address range. BIOS checks a flag byte in RAM to see whether BIOS should branch to execute special code, such as restore processor context or perform a cold boot.

After processor context is saved, software then triggers the south bridge state machine that suspends the processor in 805 by reading a register within the south bridge. The final actions of the suspend state machine triggers new run mode logic which may be implemented as a state machine. The new run mode logic updates the voltage and frequency control signals provided respectively to the voltage regulator and the frequency control logic in the CPU with the new voltage and frequency control settings and then provides a wake-up indication to trigger the resume state machine in 809. Note that voltage being supplied to the core logic is changed while core clocks are off, which mitigates risk of adverse affects of the voltage change. Alternatively, or in addition, as described further herein, core voltage may be changed while a reset is being applied to the processor. Although it is not necessary for the processor to be reset when the core voltage is

changed, it is desirable to inhibit processor operation until the core voltage has settled to the new value. The resume state machine issues a reset signal to the processor with causes the new frequency value to be applied to the clock multiplier logic. After enough time to sync up the processor's phase lock loop (PLL), e.g., approximately 1 ms or less, the resume state machine releases the reset in 813. The CPU then begins execution and jumps to the POSCCL resume routine whose address was set up previously, to restore CPU context in 815. The processor then can resume processing where it left off without impacting the application(s) in use at the time of the run mode change, other than by the time to effect the POSCCL.

Referring to Fig. 9A, the timing charts illustrate the operation of the south bridge for a POSCCL suspend and the corresponding resume operation. The transition from the on state to the suspend state is shown on the left side of Fig. 9A and the transition from the suspend state to the on state is shown on the right side of Fig. 9A. The processor, once having entered the sleep state in POSCCL, can be awakened by such events as the RTC alarm, an SMBus event, serial port, a ring indicator, the system's soft power button, an external SMI (EXTSMI), raising the systems lid, a Global Standby Timer alarm, USB activity, IRQ [1,3:15], or a General Purpose Input 1 (GPI1) assertion. In one embodiment described herein, general purpose inputs to the south bridge are used as triggering events to wake up the processor. The signals shown in Fig. 9A are described in Fig. 9B. Several signals listed in Fig. 9B may be provided as general purpose outputs (or may not be used in a specific notebook PC design).

PROGRAMMABLE LOGIC DEVICE IMPLEMENTATION

Before describing further details of various south bridges implementations which may be used in the system illustrated in Fig. 5, another approach will be described which provides a quick time to market solution. In that approach a PIIX4-compatible south bridge together with a logic device such as a programmable logic device (PLD), supply the necessary signals to the processor's (BF-pins) frequency control inputs and reprogram the core voltage power supply according to the requirements of the various run modes. The programmable logic device may be a programmable array logic (PAL) device or a programmable logic array (PLA) or other appropriate logic device.

Referring to Fig. 10, programmable logic device 101 receives the values from the jumpers 511 and provides 8 output bits, five to CPU core voltage regulator 501 and three frequency control bits BF[2:0] to CPU 503. Those bits respectively control the voltage regulator 501 and processor frequency logic on CPU 503. In addition, the south bridge 103 provides several control signals to programmable logic device 101 and receives a wake signal therefrom as described further herein. The PLD shown in Fig. 10 allows an implementation of the run mode transitions described herein without requiring design changes to other system components.

Note that if necessary, the bits supplied for voltage and frequency control can be allocated differently by giving more bits to frequency control and fewer to voltage control depending on the needs of the voltage regulator and the frequency range of operation desired. In fact, fewer than all available bits may be used for voltage or frequency or both. For example, a typical processor, like the AMD-K6 processor, needs just three bits for frequency setting. Programmable voltage regulators are available with four or five bits of control. In

many applications, the full range or precision of the voltage regulator is not needed and some control inputs to the regulator may be tied high or low. Thus, the approach herein provides flexibility by allowing varying numbers of voltage and frequency control signals to be used based on the needs of the particular system.

Referring to Fig. 11, one implementation of programmable logic device 101 is shown in greater detail. In the implementation illustrated, there are 13 inputs and 9 outputs and the design can be implemented in a standard programmable array logic (PAL) device. The signals originating from south bridge may be clocked by the real time clock (RTC) (32 kHz); therefore a high speed logic device is unnecessary.

There are eight input flip-flops 1101 connected as a serial shift register which receive a shift signal 1102 and a data in signal 1104 from south bridge 103. South bridge 103 uses one general purpose output (designated as GPO-X) as the shift signal supplied to PLD 101 and another general purpose output (designated as GPO-Y) as the data in signal to PLD 101. PLD 101 includes eight output flip-flops 1103 receiving input signals from a selector circuit 1105, which selects either one of the input flip-flops or one of the jumper settings (IBF[0:3] and IV[0:3]). On a power on reset the flip-flops are held in reset until Power Okay (PWROK) 1107 is asserted. While PWROK 1107 is asserted, south bridge 103 issues a reset to the processor.

Note that the default values of the voltage and frequency control signals represented by the jumper settings should be reapplied to the frequency and voltage control circuits if a system reset is created either by pushing the reset button or by software or by any other mechanism. Therefore, the state of the GPO bit used to drive data in 1104 must default to logic zero if that bit is also used as a select signal for select logic 1105 as shown in the illustrative embodiment depicted. If data in 1104 is zero for system reset, that assures that the default values from the jumpers settings are appropriately selected by select logic 1105. In some implementations, only a few of the PIIX4-compatible south bridge bits have this property (e.g., GPO[27:28,30]) and therefore the GPO-X and GPO-Y bits are selected from among those bits. Using one of those bits insures that the start-up jumper settings will be communicated to the voltage regulator and the processor's BF-pins during reset in the implementation shown. Other implementations would be readily apparent to one of skill in the art. Using the data in signal 207 as a multiplexer select minimizes the number of GPO pins required, although additional GPO signal(s) from the south bridge could also be used to provide the multiplexer select signal.

When the system initially powers up, the rising edge of the CPURST signal causes the default or startup voltage and frequency settings to be loaded into output registers 1103 and thus be provided to voltage regulator 501 and CPU 503. Loading the output registers will immediately set the voltage regulator to the initial value. The default frequency settings will also be output. CPURST causes the processor to loads its BF-pin inputs and sets the initial bus clock multiplier for generating the CPU core clock.

Once a need to transition to a new run mode is detected by the notebook system and prior to executing a suspend/resume sequence to change the processor voltage and bus clock multiplier, the GPO bits on the south bridge are used to load the new values into data input registers 1101. Referring to Fig. 12, the setup for the transition sequence occurs in 121 when the shift signal 1102 is used to shift in the new voltage

and frequency settings on data in signal line 1104 into input register 1101. Once that setup is completed, the suspend operation is executed which includes saving processor context and, as illustrated in Fig. 12, asserting STPCLK# and SLP#.

As described, the GPO bit 1104 used to input data into the shift registers, also steers the multiplexer 1105. Leaving the bit high during the suspend/resume sequence steers the multiplexer to supply the output of the shift registers, instead of the jumpers, to the inputs of the output registers 1103. The serial data is written to logic device writing to the appropriate GPIO ports.

Once the core voltage and frequency have been changed in the input register 1101, the south bridge is made to execute a POSCCL operation which causes the south bridge to supply a signal indicating that the suspend operation is at or near completion. As shown in Fig. 12, that is the SLP# signal although other signals could also be used. That signal indicates the end of the suspend sequence and is provided as trigger (TG#) 1109 to PLD 101. In this particular implementation, TG# is active low. PLD 101 creates an event to resume operation, i.e., wake up the processor by logically combining that trigger signal with data input 1107 in gate 1111. That is used to gate SLP# through as the wake event (active low) to south bridge input GP11# that is sensed as a wake up event resulting in a standard POSCCL resume operation, which as previously described, is useful for setting new processor frequency multiplier because it cycles the processor reset. Thus, as shown in Fig. 12, the end of the suspend operation results in the wake event. That in turn that causes the resume sequence illustrated in Fig. 12. Once the resume sequence is complete, the GPO signal 1107 used as the data input is brought low.

The assertion of CPURST during the resume sequence loads the values from the input registers 1101 into output registers 1103. The values in output registers 1103 provide the new frequency multiplier settings and the core voltage settings for the processor and voltage regulator respectively. Thus, the new voltage settings are applied to the core logic of the processor while CPU reset is asserted. The new frequency multiplier ratio BF-pin settings are sampled by the CPU on the rising edge of CPURST and latched on the falling edge of CPURST into the processor. The POSCCL resume allows time for resynchronizing the processor PLL.

Using the PLD allows an unmodified south bridge and processor to be used in a notebook system having the capability to transition between run modes.

If it is desired to keep the run mode transition logic and software as simple as possible, it is recommended that the bus clock frequency not be changed between 66 MHz and 100 MHz during the change. Since the BF settings are conventionally in ½ x steps (e.g. 2x, 2.5x, 3x, 3.5x, etc.) that limits the granularity of the processor clock frequency steps. If greater complexity is acceptable such changes can be implemented. Changing the frequency of the bus clock affects the divider ratios used to create the Peripheral Component Interconnect (PCI) and Accelerated Graphics Port (AGP) clocks. It may be necessary to put memory into a sleep or power down when changing the clock ratios.

The implementation described herein utilizes a variable voltage regulator supply such as the National Semiconductor's LM4130, whose output voltage can be controlled by the chipset logic (including any external device). It is desirable for the voltage regulator to support at least four control bits and for the output voltage to be controllable in steps of 50mV (or smaller) covering a minimum range of from 1.45 to 2.2 volts. A wider range may be desirable in some applications. The voltage regulator's control pins should be configured for default operation at the Mode 0 (battery saver) voltage level upon power-up. The chipset or other appropriate logic will then adjust the CPU voltage supply for the right run mode once the system is operational.

SOUTH BRIDGE IMPLEMENTATION

In order to provide an implementation that minimizes the number of components needed to change run modes as described herein, the south bridge can be modified to provide the logic necessary to transition between the various run modes rather than utilize an external logic device. A high level implementation of such a system is illustrated in Fig. 5.

One approach to the south bridge implementation is to incorporate the logic contained in the PLD into the south bridge. However, the logic can be simplified since the various signals needed to interface the south bridge to the PLD can be eliminated. In addition, the use of the data input signal as the multiplexer signal could be eliminated. In fact, the input registers would preferably be loaded in parallel, rather than serially thus eliminating the need for a shift signal. Once the determination is made that a new run mode is required, a register in the south bridge is provided with appropriate voltage and frequency settings.

In one implementation, the south bridge defines a new sleep type in the power management control register (see Table 1 above) using one of the combinations of bits [12:10] not presently used (e.g., 110). Those bits are variously referred to as either sleep type or suspend type. When the sleep enable bit (or suspend enable) is set to one and the sleep type bits are 110, then the system causes a transition in the notebook run modes.

Referring to Fig. 13, run mode control register 130 provides the control information necessary to transition to a new run mode. The two bit operating mode field 131 is a status field that identifies the current run mode as either high performance (00), AC power (01), battery performance (10) or battery save mode (11). Additional run modes would require additional bits. The five bit core voltage field 132 defines the control bits for the new core voltage and the four bit CPU clock frequency control bits 133 define the CPU core frequency. As discussed herein, the clock control in the CPU may be implemented as a frequency multiplier of the bus clock. The reset control bit 134 defines whether or not a reset is provided to the CPU during the run mode transition. In one embodiment, when the reset control bit is 1, the CPU is reset on a run mode transition and when the reset control bit is 0, the CPU is not reset. If a reset is provided to the CPU in order to change the operating core frequency, then it is necessary to save processor context prior to the reset and to provide a resume operation that restores processor context after the reset signal is deasserted as was discussed with relation to the POSCCL sequence.

In other embodiments, the processor may have an input signal, separate from reset, that tells the processor when to latch in the clock frequency control bits. In that case, the latch mode CMD bit 135 is asserted. In that implementation, the south bridge provides, in addition to the frequency control bits (BF[0:2]), a frequency latch control signal (CMD) 515 (see Fig. 5) indicating to the CPU when to latch in the new frequency control bits. When the frequency latch control signal CMD 515 is asserted, BF pin settings may be acquired on the assertion of the CMD 515 signal and latched on the falling edge of CMD 515. Other implementations to acquire the BF pin settings based on the latch control signal CMD, are of course possible. Use of the latch control signal CMD 515 signal provides the advantage of transitioning to a new run mode without having to provide a reset which means that processor context is not lost. Thus, the transition is significantly less time consuming than the POSCCL suspend and resume operation.

If the CPU reset signal is not used to indicate a frequency change, once the need to change run modes is detected, software loads up the appropriate values in register 130 and stops the processor clocks. The clocks can be stopped on the processor using the STPCLK# signal as described previously. The new voltage and frequency control bits are output by the south bridge and the frequency control latch signal CMD 515 is asserted to indicate that updated frequency control signals are available. The CPU samples the frequency control bits (BF pins) on the assertion of the latch control signal CMD 515 and latches in the new value on deassertion of the signal. Hardware in the south bridge maintains the latch control signal CMD 515 asserted for a sufficient length of time for the processor PLL to stabilize. Once the processor PLL is stabilized, the south bridge hardware deasserts the STPCLK# signal and the processor resumes operation without the application or user knowing about the run mode transition. The time to make the transition is less than, e.g., 100 μsec.

If reset is used to indicate a change in frequency, to avoid requiring changes to the processor, a sequence similar to POSCCL illustrated in Fig. 9A may be used, with new control signals for voltage and frequency being provided between the latter part of the power on suspend (POS) sequence and the assertion of reset at 91 in Fig. 9A.

The transition sequence from one run mode to another is initiated when an interrupt (e.g. SCI) is generated because of an operating mode event. The event may be because the notebook computer is plugged into or removed from a docking station, a port replicator or any other device than can remove large amount of thermal energy from the notebook; or when AC power is supplied or removed or when the battery is running low or any other event that should result in a change in the operating mode.

The system management software, once the event is detected programs the operating mode control register 130 with the new core voltage and the new clock frequency control bits. The software then sets the sleep type bits to 110 as well as the sleep enable bit. The reading of special register LVL3 in south bridge 505 starts the hardware state machine that performs the POSCCL sequence. The use of operating mode control register 130 may be advantageously employed with a south bridge only implementation and also with the PLD implementation previously described.

As an alternative to starting the change in run mode sequence by reading the LVL3 register, e.g., in a non-ACPI environment, writing (or reading) register 130 may be used as a trigger to start the control logic to implement the run mode change. Such a trigger may be used when the run mode change utilizes either the latch control signal CMD or the reset signal. In such a case, saving processor context would have to be completed before writing the register if a reset is utilized. If latch control signal CMD 515 is used, then that signal may be strobed, e.g., for one PCI clock, to indicate that the BIF signals are valid after register 130 is written.

Referring again to Fig. 5, note that the south bridge still needs to ensure that the default value from the jumper settings 511 are provided to CPU 503 and the voltage regulator 501. Referring to Fig. 14, that can be accomplished by providing, in a similar fashion to the PLD implementation, a multiplexer 141 that selects between the values in the control register 130 and jumper settings 511 according to select signal 142 supplied from control logic 144. Control logic 144 also contains the necessary suspend and resume state machines to implement the suspend and resume sequences illustrated, e.g., in Fig. 9A. Multiplexer 141 supplies output register 143 with values from jumper settings 511 which selects the default jumper settings in response to reset (e.g. a power on reset or other hard or soft reset). In addition, output register 143 is loaded after clocks are stopped and before supplying either a reset or CMD signal to load in new frequency settings or on the rising edge of reset of CMD assuming the falling edge causes a new frequency setting to be latched.

Referring again to Fig. 13, control bit 136 defines whether or not the clock generator 507 is disabled or not during the run mode transition. Along with using the STPCLK# signal from the south bridge to disable the core CPU clocks, it is also possible to disable the clocks supplied to the CPU at clock generator 507 (as in POSCCL). The clock frequency control bits are then updated, and the clocks are turned back on at clock generator 507. Note that clock generator 507 may be integrated with other system components. It is desirable to have a clock generator whose outputs can be enabled by the south bridge's GPO control bits. The clock generator may have multiple PLL cells to support the various clock frequencies desired in any particular design, e.g., supporting the CPU at 100MHz or 66MHz, serial devices at 24KHz and 48KHz, and PCI devices at 33MHz.

Referring to Fig. 15, the flow chart illustrates the overall operation of the south bridge incorporating hardware to cause run mode changes. When an event is detected that results in a run mode change, such as an additional power source becoming available as described previously herein, software sets the required values for register 130 in 1501. In 1503, a determination is made as to whether or not reset control bit 134 is set in register 130. If so, then it is necessary to save processor context in 1504. That determines the exact sleep type that will be used by the south bridge. If the reset control bit 134 is not being used, that presumes that the latch command bit 135 is set and the step of saving processor context 1504 may be skipped. Of course, only one bit needs to be used to indicate whether to use a reset or latch command signal. If accessing register 130 is used as a trigger to start a change in run mode, then context may need to be saved prior to such an access. Once processor context is saved in 1504 or if a reset is not being used, then state machine in south bridge can take over to perform following the steps to affect the run mode change.

In 1505, the software executes a read of register LVL3 which starts a state machine sequence to effect the run mode change. In 1507, south bridge asserts STPCLK#. The state machine or other appropriate hardware and/or software waits for Stop Grant special bus cycle to be received which is an indication from the processor that it has turned off its internal core clocks. Note that the Stop Grant acknowledgement can be performed without monitoring of the CPU bus cycles by simply waiting a predetermined time which is more than the maximum time that could be taken to reach the Stop Grant state by the processor.

In any case, once Stop Grant is acknowledged or the predetermined time limit is reached and it is assumed that Stop Grant state exists in the processor, it is determined if the stop clock bit 136 in control register 130 is set. If the bit is set, then in 1513, the clock output to the CPU from clock generator 507 is turned off using the stop clock line (enable 509). Then in 1515, new voltage and frequency control settings are applied to the voltage regulator and the BF pins, respectively. In 1517, it is determined whether the stop clock bit (enable 509) is asserted. If so, in 1519, clock generator 507 is enabled to output the clock to the CPU and sufficient time is provided for the PLL to stabilize. In 1521, it is determined whether reset is being used to latch in the updated frequency control values. If so, then in 1523, CPU reset is strobed to latch in the new frequency control bits. If reset has not been used, then the CMD signal 515 is strobed by the south bridge to cause the processor to latch in the new frequency without a reset. In 1527, the STPCLK# signal is deasserted. That causes the CPU to resume supplying CPU core clocks. The processor then resumes executing 1529. Note that if context was lost because reset was used, context is restored at this point. That is accomplished under software control and no longer under hardware (e.g. state machine) control.

Use of register 130 provides an embodiment in which the use of either reset or the latch control signal CMD is selectable, and turning off the clocks external to the processor is also selectable. In other embodiments, the south bridge (or other suitable integrated circuit) may not provide options. In other words, in other embodiments, reset may always be used or the latch control command signal may always be used. Further, the change run mode sequence may always turn off clocks external to the processor (as well as internally) or may always leave such clocks running.

If the notebook PC uses both a 66 MHz and 100 MHz bus clock frequency, the clock generator may have multiple PLL cells to be able to slew the CPU clock input while maintaining other system required frequencies constant. The clock generator would be able to slew the CPU frequency across the desired range while keeping the rate of change in the CPU clock frequency within the jitter specifications of the CPU to prevent the clock multiplier circuitry within the processor from loosing lock.

THERMAL MANAGEMENT

Adjusting processor operating frequency and voltages optimizes power consumption and heat generation and dissipation according to environment. In addition thermal management capability is required. ACPI has two built-in schemes for thermal management, one passive and one active. The passive scheme relies on throttling down the processor to generate less heat while the active scheme uses a cooling device like a fan to remove heat from the processor and system. A suitable thermal sensor measures the processor

temperature for passive and active schemes. The thermal design is based upon one or more thermal zones. For each zone, up to three thermal thresholds can be defined:

There are two methodologies used today for thermal monitoring: a total system monitoring approach and a CPU-only monitoring. The total board monitoring philosophy provides a means of measuring all voltages (e.g., CPU core, CPU I/O, 3.3 V, 5V 12V, -12V -5), fan rotation speed, temperature of the CPU, and temperature of the board. Devices like the National LM78 may be used for this system monitoring approach. The CPU-only monitoring methodology can use the National LM75. The LM78 has an open collector output that is used to create an interrupt when the processor's temperature is above acceptable limits or when the temperature has changed by a certain amount. The System Management Bus (SMBus) may be used to program the over-temperature and temperature hysterisis values within the device. The SM Bus is a slow 2 bit serial bus that is used for communicating with monitoring devices like thermal sensors, chassis intrusion alert sensors and to provide fan speed control. Conventional south bridges have an SMBus interface so system software can talk to (setup and control) and read from (accept data from) such devices. The National LM77 device has separate outputs to indicate over temperature or if temperature is above or below certain set points, which provides added reliability. If ACPI should fail to respond to the set point interrupt, the output indicating over-temperature can shutdown the system through hardware.

ACPI maintains temperature set-points that are compared with the processor temperature. When the temperature exceeds the set-points, action is taken to reduce heat dissipation by the processor (passive method) or to expel the heat from the system (active method). The necessary registers and interface are generally contained in the PIIX4 south bridges. Most new south bridges are PIIX4 compatible. That helps reduce the effort necessary to adapt ACPI BIOS and operating system code for different company's chipsets.

ACPI maintains set-points for processor thermal management. One is a fail safe set-point that will initiate a shutdown if the processor becomes too hot. Other set-points are associated with ACPI's "Active" and "Passive" cooling methods. Either or both methods can be incorporated into a notebook design incorporating the run modes described herein.

In active cooling mode, ACPI turns on and off a cooling device based on temperature reports generated by a sensor placed on the processor heat sink or directly on the processor. When the temperature sensor senses a change in temperature (usually 5 degrees) it reports a new temperature to ACPI's thermal management. If the temperature is above a limit provided in an ACPI table, the cooling device is switched on. When the temperature sensor reports a drop in temperature that places the temperature below another table value, the cooling device is switched off.

Note that these thresholds are programmable, and software is allowed to dynamically adjust the thresholds for optimum results. For example, when the active cooling threshold is crossed, the system software may start the system fan at a low speed, and reprogram the active cooling threshold at a higher temperature. If the system temperature continues to rise, it will eventually cross the new active cooling

threshold. In response, the fan speed can be increased, and if appropriate, the active cooling threshold can be set again at an even higher point.

Generally, the cooling device is a miniature fan placed on or near the processor's heat sink. Running the fan circulates air from outside the notebook PC's case to cool the processor more than can be done by conduction and convection alone. The active cooling system may also be more sophisticated. When docked, an external fan in the port replicator or docking station can circulate more air. Other technologies that may be used include heat pipes, large thermal dissipation plates or refrigeration devices like the Peltier junction devices that have been used to cool desktop processors in the past.

Mobile systems may use different active cooling device and temperature set points in the ACPI tables for each mode of operation (AC/battery and docked/undocked). Since active devices consume power themselves, it is recommended that these devices receive somewhat limited use in Run Mode 0, the battery life mode. Active cooling is a good solution when the system is running from an external power source such as AC-line adapter, car adapter or airplane adapter.

In passive cooling mode, ACPI power management dynamically reduces or increases the "speed" of the processor as necessary to maintain the processor operating temperature at a safe level. Obviously, "throttling" down the processor reduces performance, but even operating at somewhat reduced speeds, today's processors can provide adequate performance for running applications such as word processing. The throttling method used in most notebooks, directly supported by ACPI and PIIX4-compatible south bridges reduces the average processor speed by alternately starting and stopping the processor using the STPCLK# (stop clock) input of the processor.

Most throttling implementations use three bit granularity. The duty cycle ranges from 12.5% to 100% in 12.5% steps. (Note that zero is not an acceptable value.) When temperature rises to an upper limit throttling is initiated at a level determined by an ACPI table value until the temperature falls below a lower limit.

The stopping and starting of the processor is not noticeable to the user. The reason is that the frequency at which the start/stop action takes place is faster than a person can perceive. PIIX4-compatible chipsets use the real time clock to drive a three bit counter that determines the duty cycle.

Mobile systems operating in the extended battery life mode (Run Mode 0) should use clock throttling if necessary, to reduce heat generation. Mobile systems should have sufficient passive cooling when operating in Run Mode 0 with a room temperature of 25 degrees C that throttling is rarely needed. In elevated temperature environments, the processor can be throttled down more frequently.

In order for a notebook PC to achieve desktop performance levels when docked, it requires additional thermal assistance from e.g., the docking station (or other solutions providing additional cooling capability and external power). The additional power dissipated through such solutions allows for a higher CPU power budget, thus permitting the processor to operate at desktop power and performance levels.

Referring to Fig. 16, a docking station solution with the notebook is illustrated. Fig. 16 shows an exemplary system that uses a Peltier device 163 in the docking station 162 to cool probe 165 that inserts into the back of the notebook computer 161 when the notebook system is docked in docking station 162. Probe 165 makes contact with the processor's heat sink and conducts heat away. The probe 165 is cooled by the Peltier device 163 which requires several watts of power available from the docking station's AC operated power supply.

Other options are available such as heat pipe technology. Heat pipes are generally made of tubes with water or other coolant under low pressure. When one end is heated by the processor, the coolant absorbs heat when it vaporizes. The vapor moves to the other end of the pipe where it is cooled by a heat sink. The vapor condenses as it cools and is moved by capillary action through a wick back to the starting point. Heat pipes can be made many different ways and can be connected to large surfaces like a metal case to dissipate heat. Thus, notebook docking station solutions may include fans, heat sinks, and heat pipes to conduct heat away from the notebook computer when docked to allow for increased performance desired in run mode 3.

Note that if thermal conduction is used, no user accessible part may rise above 50° C. No commonly contacted area should be uncomfortable to touch. Note also that while forced air from the docking station may increase processor cooling, it may also be noisy and cause a dust problem.

The mechanical and electrical design should support the necessary sensing so the notebook computer can detect when the computer is being docked, is currently docked and when it is being undocked, when auxiliary power and/or auxiliary cooling is available and or when user defined operational states have been modified. In addition the notebook PC should detect when AC power is first applied, when AC power is present, and when AC power has been removed. The design should support detecting all standard notebook states such as power and reset buttons, when primary battery is present, the remaining capacity of battery (i.e. Smart Battery), the battery charging state, whether a secondary battery is present in an option bay, the remaining capacity of that battery, and that battery charging state. It should detect charging states include fast charge, trickle charge, battery fault (e.g., shorted), and not charging. It should detect when suspend button (or key combination) is pressed to suspend or wake the system, when the cover is closed or opened (e.g. option to suspend and wake or just to shutdown backlight).

The system should also sense processor case temperature and at least two set points for temperature "alarms" (one for turning on a cooling device like a fan when the temperature is near the safe upper limit for operation, and one for immediate protective action when the temperature reaches a critical upper limit for preventing damage to the processor), If a fan is used for auxiliary cooling, confirmation that the fan is running is provided (it is desirable that the speed of the fan can be sensed). The sensing capability described above is known in the art and is not further described herein.

As described herein, a notebook computer dynamically adapts to its environment to provide improved power and thermal management and optimizes its performance for its environment. Note that the description of the invention set forth herein is illustrative, and is not intended to limit the scope of the invention as set

forth in the following claims. For instance, while this invention has been described with relation to a class of mobile computers referred to herein as notebooks (which may also be referred to as laptops or portable computers), the teachings herein may also be utilized in other portable computing devices such as personal digital assistants, (PDAs) which are handheld devices that typically combine computing, telephone/fax, and networking features or in other small form factor computing and/or communication equipment where such run modes may prove useful. Other variations and modifications of the embodiments disclosed herein, may be made based on the description set forth herein, without departing from the scope and spirit of the invention as set forth in the following claims.

WHAT IS CLAIMED IS:

1. A method of controlling the power consumption of an integrated circuit in an electronic system, comprising:

operating the integrated circuit at a first voltage and at a first frequency;

detecting a change in at least one of a plurality of operating characteristics in the electronic system;

in response to detecting the change, stopping clocks running on at least a substantial portion of the integrated circuit;

supplying updated frequency control information to clock control logic in response to the change; and

restarting the clocks to operate the integrated circuit at a second clock frequency corresponding to the updated frequency control information.

2. The method as recited in claim 1 further comprising:

supplying updated voltage control information to a voltage control circuit in response to the change; and

operating the integrated circuit at a second voltage corresponding to the updated voltage control information.

3. The method as recited in claim 2 wherein the clock control logic is disposed on the integrated circuit and at least some clock signals are active on the integrated circuit while the clocks are stopped on the substantial portion of the integrated circuit and wherein clocks are stopped on the substantial portion of the integrated circuit according to a clock control signal supplied to the integrated circuit.

4. The method as recited in claim 2 wherein the frequency control information is provided as clock multiplier information to clock multiplier logic on the integrated circuit.

5. The method as recited in claim in any of claims 1 through 4 wherein the integrated circuit is a processor and wherein the electronic system is a notebook computer system.

6. The method as recited in any of claim 1 through 4 wherein the operating characteristics include a power source characteristic and a thermal environment.

7. The method as recited in any of claims 1 through 4 wherein the operating characteristics include user selected operating parameters.

8. The method as recited in claim 6 wherein the power source characteristic includes the presence of external power and wherein the thermal environment includes the availability of auxiliary cooling.

9. The method as recited in any of claims 1 through 4 further comprising saving processor context prior to stopping the clocks and restoring the processor context after restarting the clocks.

10. A computer system comprising:

an integrated circuit including a first logic portion, the first logic portion coupled to receive a first clock and a first voltage;

a programmable voltage regulator circuit supplying a variable voltage level for the first voltage according to voltage control signals provided to the voltage regulator circuit; and

a clock control circuit operable to generate the first clock at a frequency determined according to frequency control signals; and

a control circuit coupled to receive an indication of a change in at least one of a plurality of operating characteristics in the computer system, the control circuit responsive to the change in operating characteristics to provide the voltage control signals and the frequency control signals indicating a new voltage value for the first voltage and a new frequency for the first clock, the new voltage value and the new frequency corresponding to the change in the operating characteristics.

11. The computer system as recited in claim 10 wherein the operating characteristics include presence of external power and availability of auxiliary cooling and include user selected operating parameters selecting between a battery performance mode and a battery save mode for the computer system.

12. The computer system as recited in claim 10 wherein the control circuit is disposed on a second integrated circuit and the second integrated circuit is coupled to supply a clock stop signal to the integrated circuit, the clock stop signal being asserted in response to the change in operating characteristics, the clock stop signal causing the first clock to be stopped on the integrated circuit and wherein the clock stop signal is deasserted after signals indicative of new voltage and frequency control signals have been provided to the voltage regulator and the frequency control logic, respectively.

State	Definition	CPU State	Remarks
G0	Working	C0–C3	Computer is operating. The power of individual devices is actively managed. Mobile systems add four sub–states for power management.
G1	Sleeping	C3 or Power Off	Computer operation is suspended and the processor is in its lowest power saving state. See S1 through S4 states.
G2	Soft Off	Power Off	The computer is in a "soft/off" state. Pressing the soft–power button will initialize the system and start the boot sequence.
G3	Mech Off	Power Off	The computer is mechanically off. Applying power to the system and pressing the soft–power button will initialize and boot the system.

FIG. 1A

State	CPU State	CPU Clk	Memory	I/O	Remarks
C0	Working	Core Clock Actively Throttled	Working	Working	Individual devices are being actively managed. Mobile adds 4 sub–states for power management.
C1	HALT	Core Clock Stopped, Clock Generator working	Napping in D2 State	Can Send Accesses To Memory	Processor operation is suspended. Very low latency for restoring operation. Software can operate without consideration for wake up time.
C2	STPCLK# Granted	Core Clock Stopped, Clock Generator working	Napping in D2 State	Can Send Accesses To Memory	Processor suspended. Better power saving than C1, but longer latency for restoring operation.
C3	STPCLK# Granted	CPU Clock Generator Output May be Disabled	Napping or Powered Down	No I/O Cycles Allowed	Processor suspended. Lower power but longer latency than C2. Cache flushed. No snooping.

FIG. 1B

State	CPU State	CPU Clock	Mem State	I/O	Remarks
S0	Working	Working	Working	Working	System fully operational
S1	Deep Sleep or C3 State	Clock Generator Output Disabled	Clock Stopped, Self Refresh	No I/O Cycles Allowed	Highest power consumption and lowest wake latency sleep state. System context is maintained. The processor, Memory, PCI, AGP clocks stopped. Full operation can be restored within a few milliseconds.
S2	Powered Down	Clock Generator Output Disabled	Clock Stopped, Self Refresh	No I/O Cycles Allowed	Longer wake latency compared to S1, because the processor & cache state is lost. Processor power may be removed. This mode is optional for mobile.
S3 (Suspend To RAM)	Powered Down	Powered Down	Clock Stopped, Self Refresh	Powered Down (except event gen/monitoring circuitry)	A longer latency state than S2, the processor and chipset state are lost, but memory is maintained. To wake the processor and cache state are restored from memory.
S4 (Suspend To Disk)	Powered Down	Powered Down	Powered Down	Powered Down (except event gen/monitoring circuitry)	Lowest power consumption and the longest latency for waking. The system context has been stored, generally on the hard drive.
S5 (Soft Off)	Powered Down	Powered Down	Powered Down	Powered Down (except event gen/monitoring circuitry)	Fully off. A full reboot is necessary when the power button is pressed (a.k.a. "soft off" since the soft power button is used to start the system).

FIG. 1C

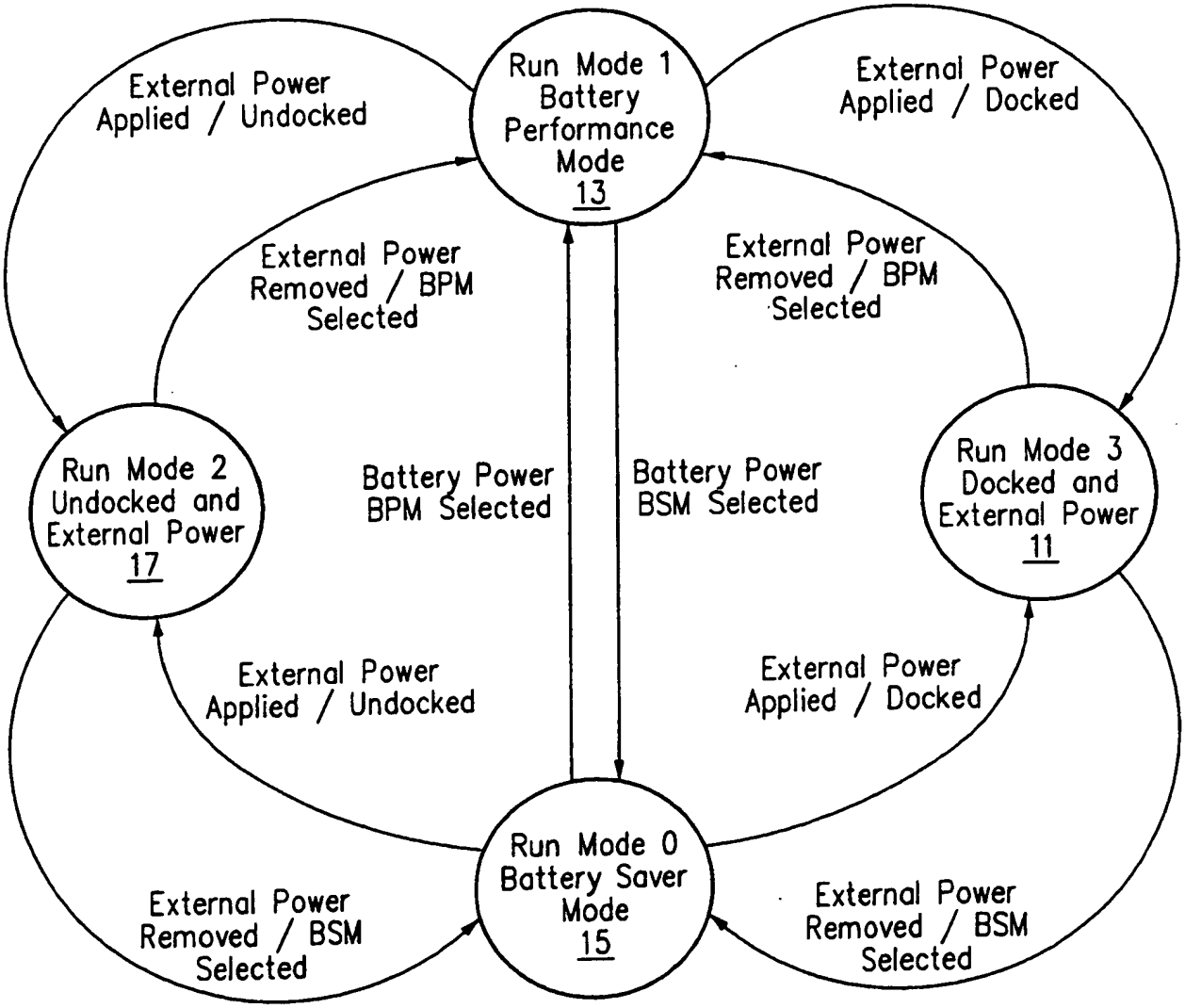

Run Mode 1
Battery
Performance
Mode
13
External Power
Applied / Undocked
External Power
Applied / Docked
External Power
Removed / BPM
Selected
External Power
Removed / BPM
Selected
Run Mode 2
Undocked and
External Power
17
Battery Power
BPM Selected
Battery Power
BSM Selected
Run Mode 3
Docked and
External Power
11
External Power
Applied / Undocked
External Power
Applied / Docked
Run Mode 0
Battery Saver
Mode
15
External Power
Removed / BSM
Selected
External Power
Removed / BSM
Selected

FIG. 2

Run Mode 3	External Power	Docked	Thermal assistance from docking station; CPU runs at desktop frequencies and power	ACPI "Active" cooling from Docking Station	TDP_{MAX} < 20 W
Run Mode 2	External Power	Un-docked	CPU runs at moderate frequency & voltage, maximizing performance within thermal constraints	ACPI "Active" cooling fan	TDP_{MAX} < 11 W
Run Mode 1	Battery Power	Un-docked	CPU runs at moderate frequency & voltage, maximizing performance with moderate battery life	ACPI "Active" cooling fan	TDP_{MAX} < 11 W
Run Mode 0	Battery Power	Un-docked	Runs at lower frequencies and voltages, maximizing battery life	ACPI "Passive" cooling	TDP_{MAX} < 6 W

FIG. 3A

Mode	CPU Voltage	CPU Frequency	Bus Multiplier	Bus Frequency	PCI Frequency
Run Mode 3 Ext Power/ Docked	2.2 Volts	400 MHz	4X	100 MHz	33.3 MHz
Run Mode 2 Ext Power/	1.8 Volts	300 MHz	3X	100 MHz	33.3 MHz
Run Mode 1 Battery Performance	1.8 Volts	300 MHz	3X	100 MHz	33.3 MHz
Run Mode 0 Battery Life Mode	1.6 Volts	200 MHz	2X	100 MHz	33.3 MHz

FIG. 3B

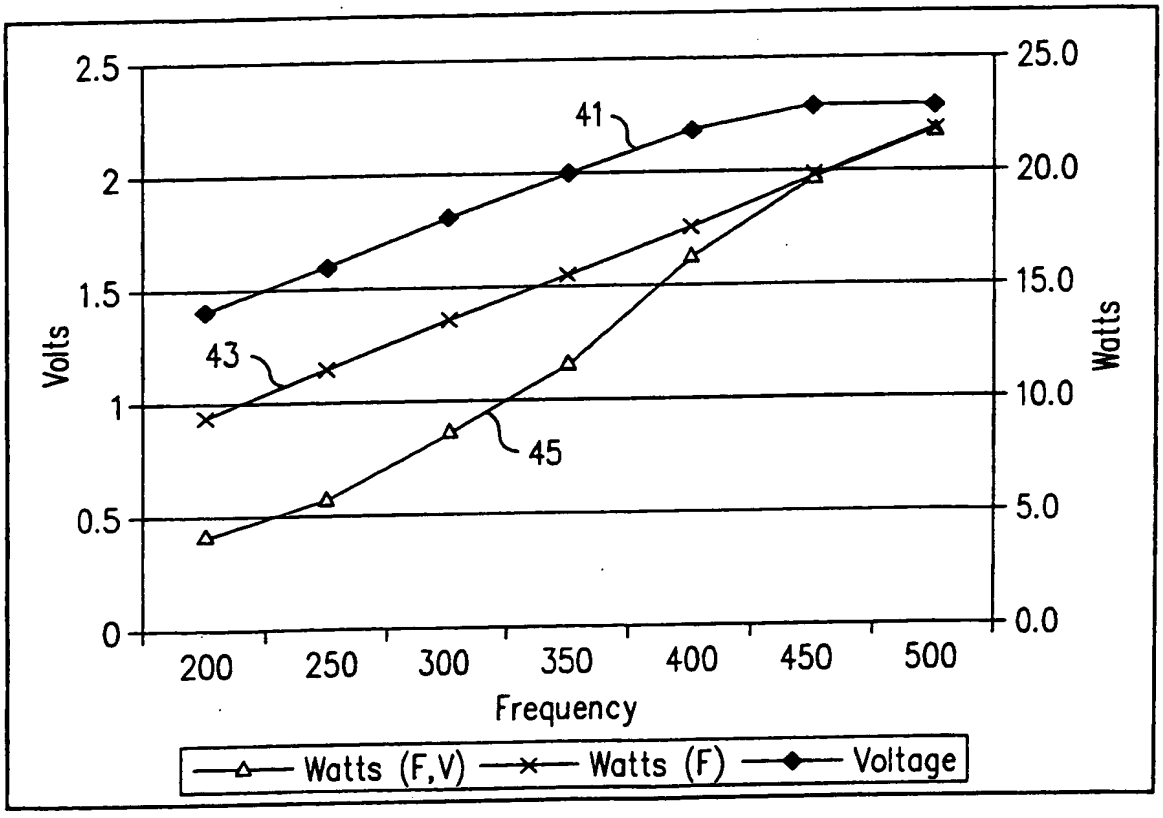

2.5
2
1.5
1
0.5
0
Volts
25.0
20.0
15.0
10.0
5.0
0.0
Watts
41
43
45
200
250
300
350
400
450
500
Frequency
Watts (F,V)
Watts (F)
Voltage

FIG. 4

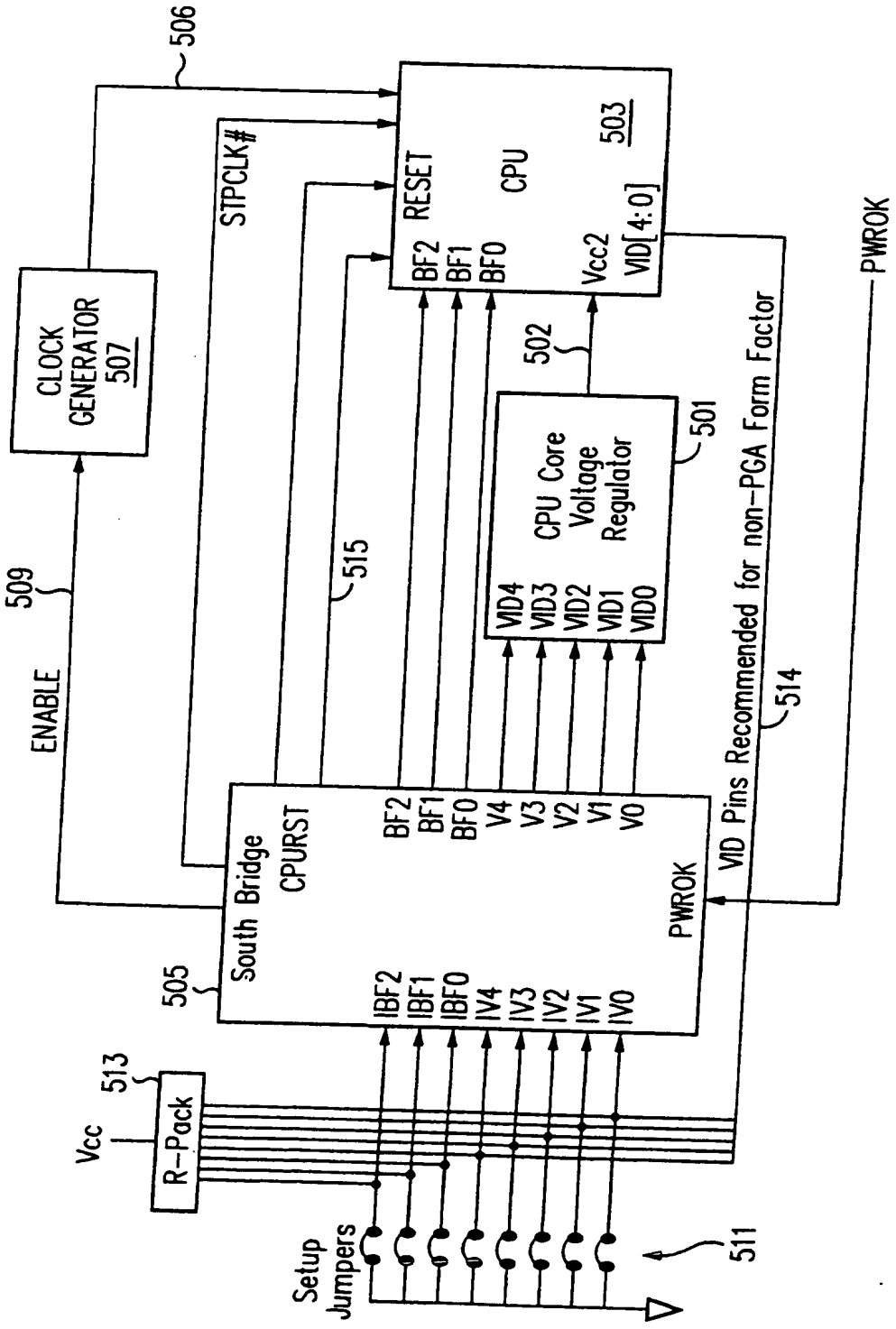

506
CLOCK GENERATOR 507
STPCLK#
RESET
CPU
503
BF2
BF1
BF0
Vcc2
VID[4:0]
PWROK
502
501
CPU Core Voltage Regulator
VID4
VID3
VID2
VID1
VID0
509
ENABLE
515
514
VID Pins Recommended for non-PGA Form Factor
South Bridge
CPURST
BF2
BF1
BF0
V4
V3
V2
V1
V0
PWROK
505
IBF2
IBF1
IBF0
IV4
IV3
IV2
IV1
IV0
513
Vcc
R-Pack
Setup Jumpers
511

FIG. 5

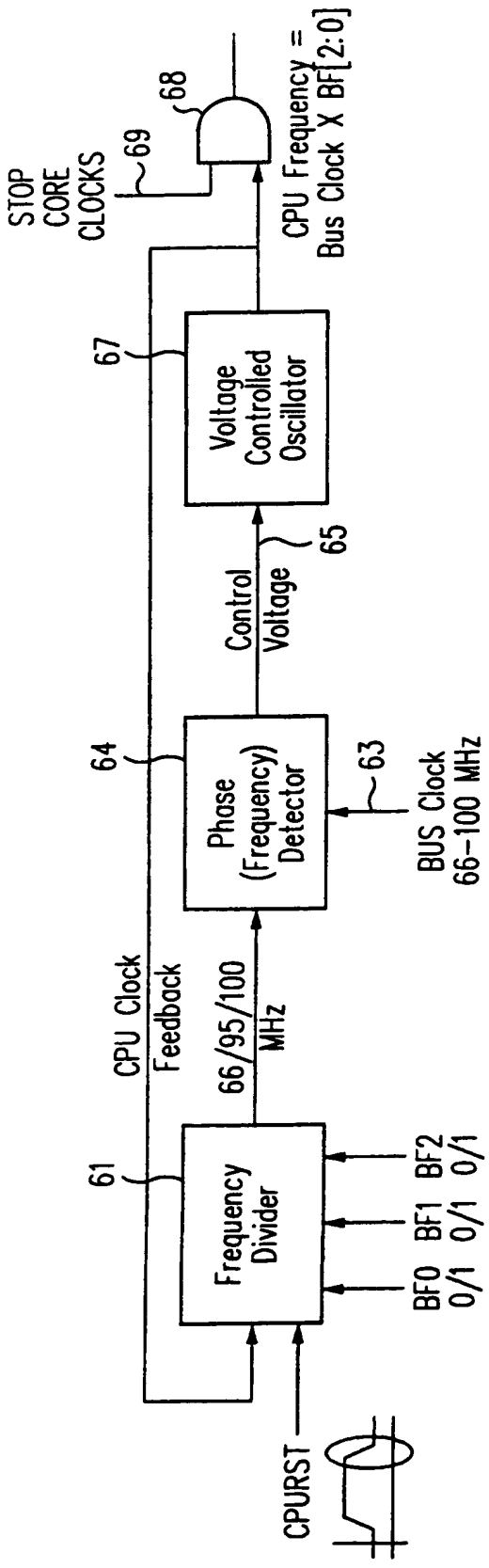

CPURST
Frequency Divider
61
BF0 BF1 BF2
0/1 0/1 0/1
CPU Clock Feedback
66/95/100 MHz
Phase (Frequency) Detector
64
63
BUS Clock 66–100 MHz
Control Voltage
65
Voltage Controlled Oscillator
67
68
STOP CORE CLOCKS
69
CPU Frequency = Bus Clock X BF[2:0]

FIG. 6

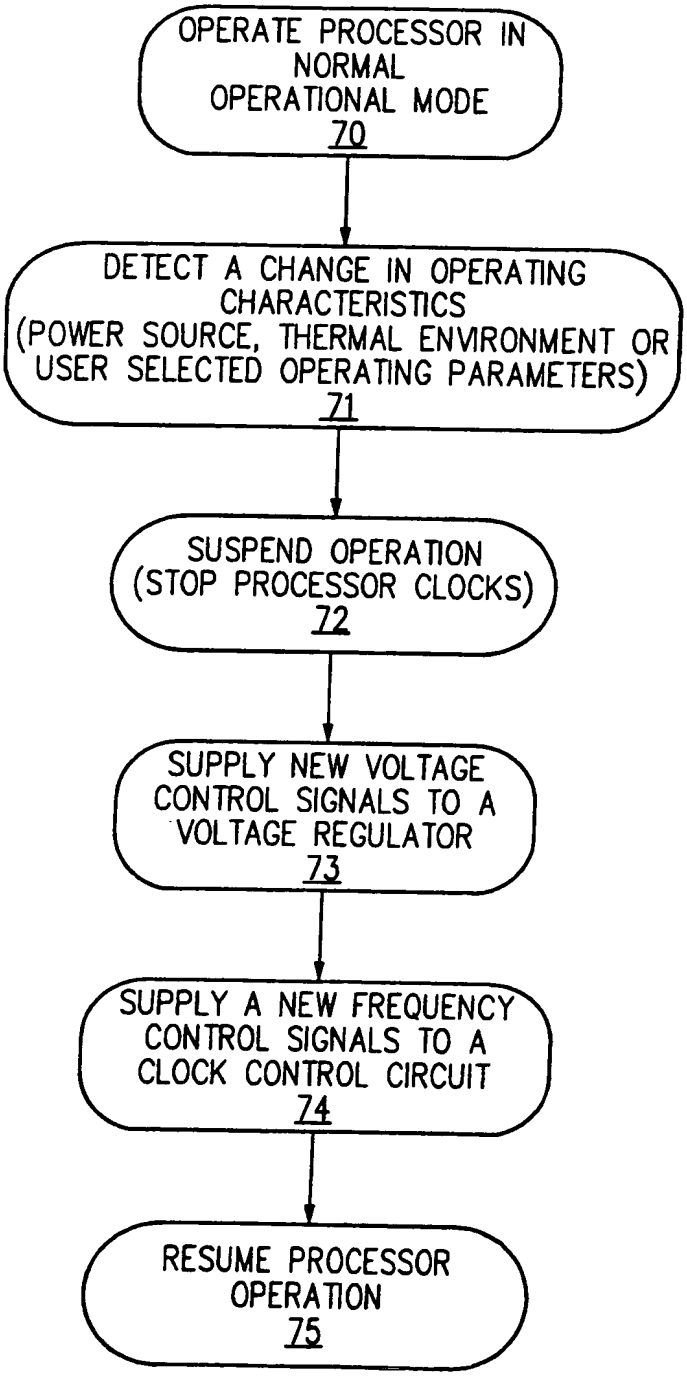

OPERATE PROCESSOR IN NORMAL OPERATIONAL MODE
70
DETECT A CHANGE IN OPERATING CHARACTERISTICS (POWER SOURCE, THERMAL ENVIRONMENT OR USER SELECTED OPERATING PARAMETERS)
71
SUSPEND OPERATION (STOP PROCESSOR CLOCKS)
72
SUPPLY NEW VOLTAGE CONTROL SIGNALS TO A VOLTAGE REGULATOR
73
SUPPLY A NEW FREQUENCY CONTROL SIGNALS TO A CLOCK CONTROL CIRCUIT
74
RESUME PROCESSOR OPERATION
75

FIG. 7

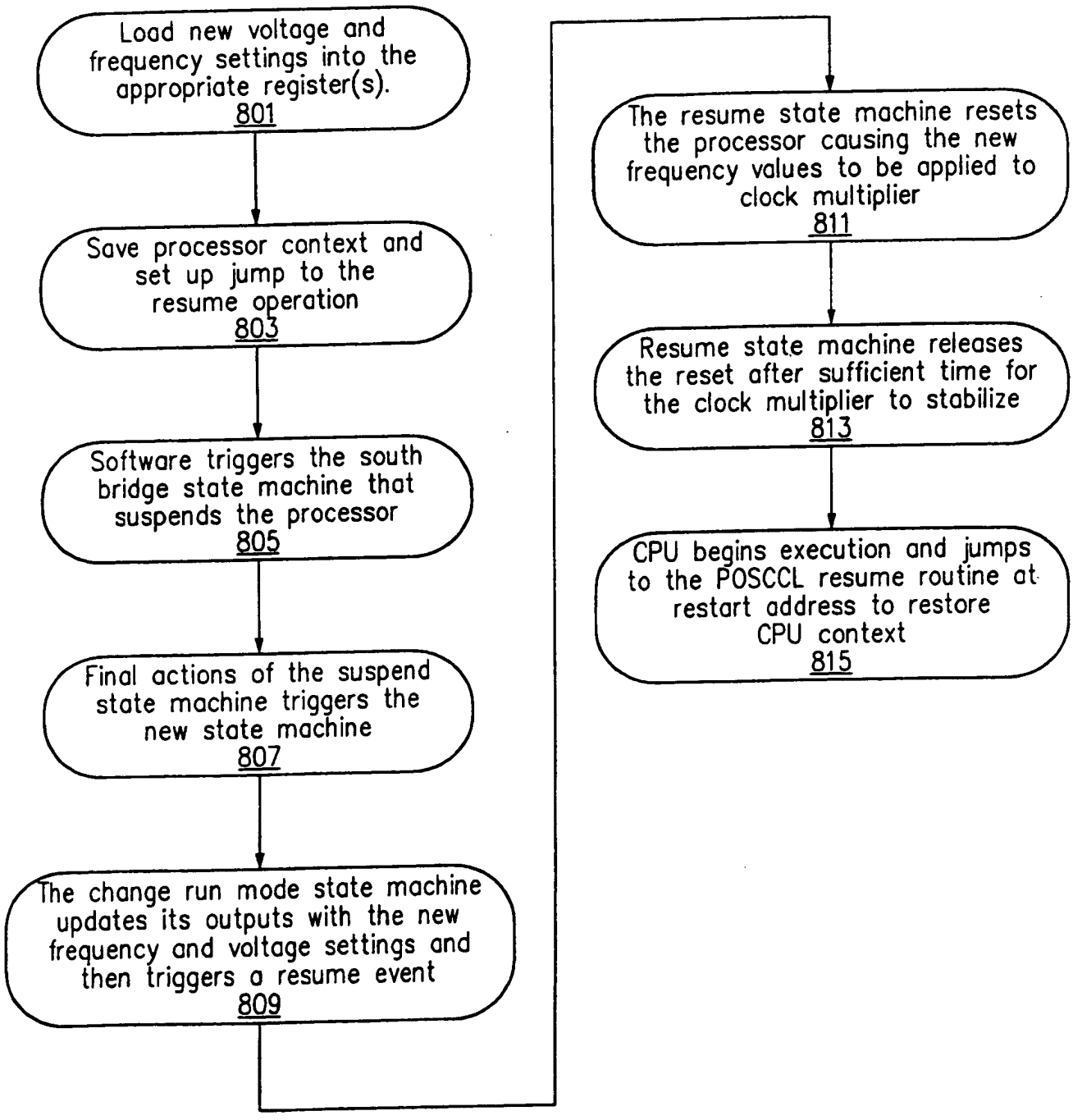

Load new voltage and frequency settings into the appropriate register(s).
801
Save processor context and set up jump to the resume operation
803
Software triggers the south bridge state machine that suspends the processor
805
Final actions of the suspend state machine triggers the new state machine
807
The change run mode state machine updates its outputs with the new frequency and voltage settings and then triggers a resume event
809
The resume state machine resets the processor causing the new frequency values to be applied to clock multiplier
811
Resume state machine releases the reset after sufficient time for the clock multiplier to stabilize
813
CPU begins execution and jumps to the POSCCL resume routine at restart address to restore CPU context
815

FIG. 8

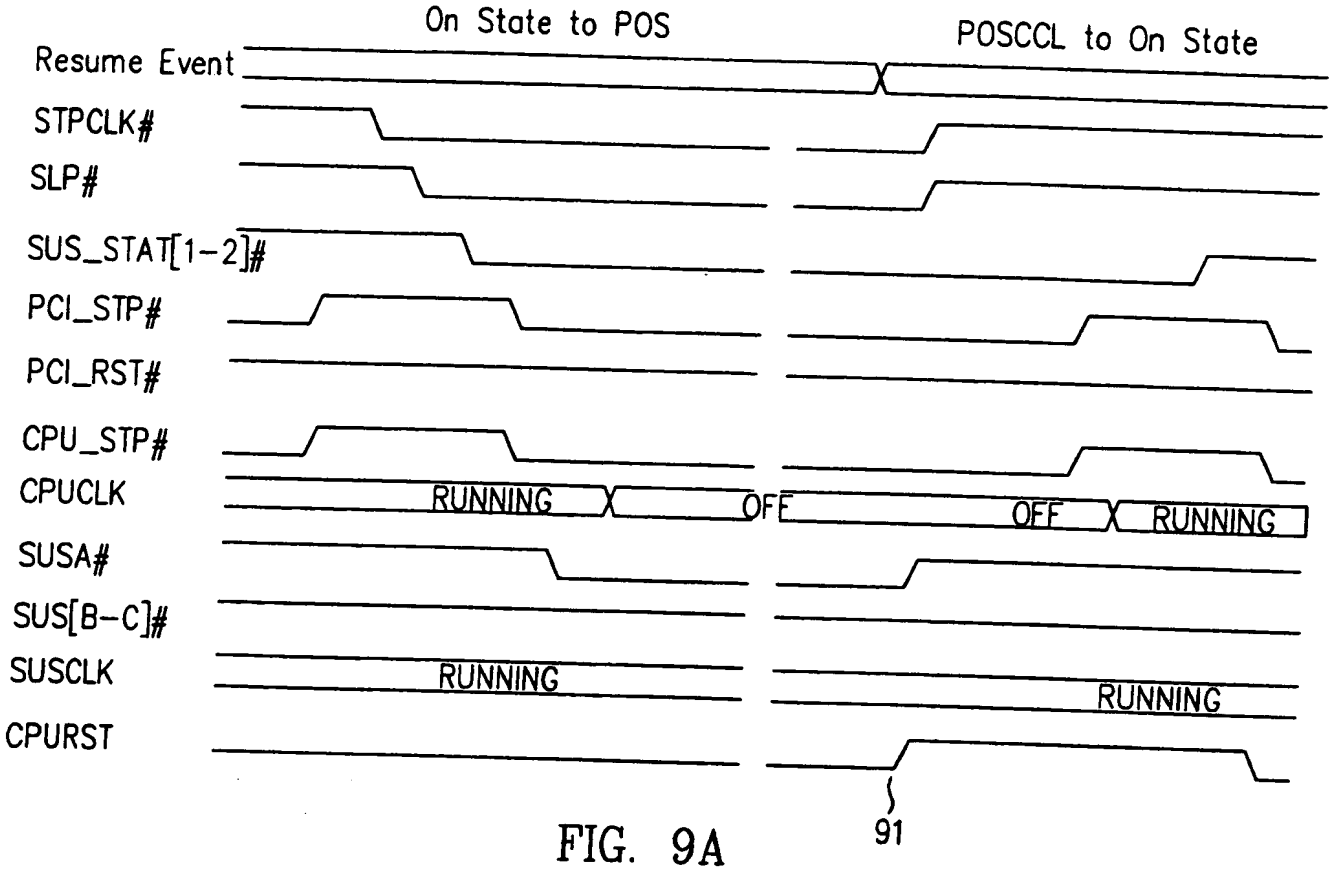


FIG. 9A

Signal	Signal Description	Signal Destination	Suspend Type	Type of Output
SLP#	Sleep	To CPU	Stop Clock, POS, STR, STD	
SUS_STAT1#	Suspend Status 1	To North Bridge	Stop Clock, POS, STR, STD	GPO
SUS_STAT2#	Suspend Status 2	To System Peripherals	POS, STR, STD	GPO
PCI_STP#	PCI Clock Stop	To Clock Synthesizer	POS, STR, STD	GPO
CPU_STP#	CPU Clock Stop	To Clock Synthesizer	POS	GPO
SUSA#	Suspend Plane A Control	Primary Power Plane	POS, STR, STD	
SUSB#	Suspend Plane B Control	Secondary Power Plane	STR, STD	GPO
SUSC#	Suspend Plane C Control	Tertiary Power Plane	STD	GPO
SUSCLK	Suspend Clock	To North Bridge (32 Hz)	STD	
CPURST	CPU Reset	To CPU	POS, STR, STD	

FIG. 9B

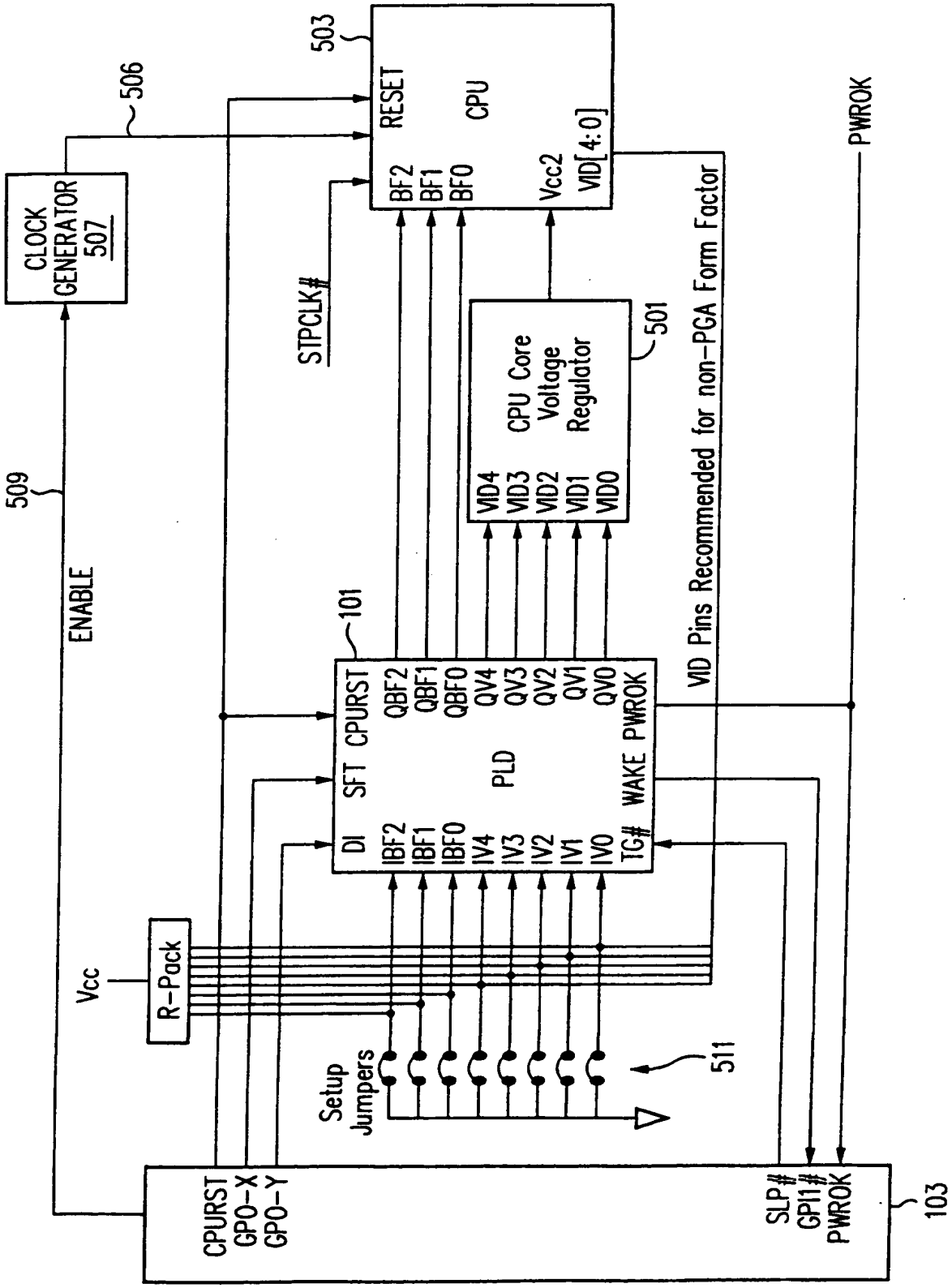

503
RESET
CPU
BF2
BF1
BF0
Vcc2
VID[4:0]
506
CLOCK GENERATOR 507
STPCLK#
501
CPU Core Voltage Regulator
VID4
VID3
VID2
VID1
VID0
PWROK
VID Pins Recommended for non-PGA Form Factor
509
ENABLE
101
CPURST
QBF2
QBF1
QBF0
QV4
QV3
QV2
QV1
QV0
SFT
PLD
WAKE
PWROK
DI
IBF2
IBF1
IBF0
IV4
IV3
IV2
IV1
IV0
TG#
Vcc
R-Pack
Setup Jumpers
511
CPURST
GPO-X
GPO-Y
SLP#
GPI1#
PWROK
103

FIG. 10

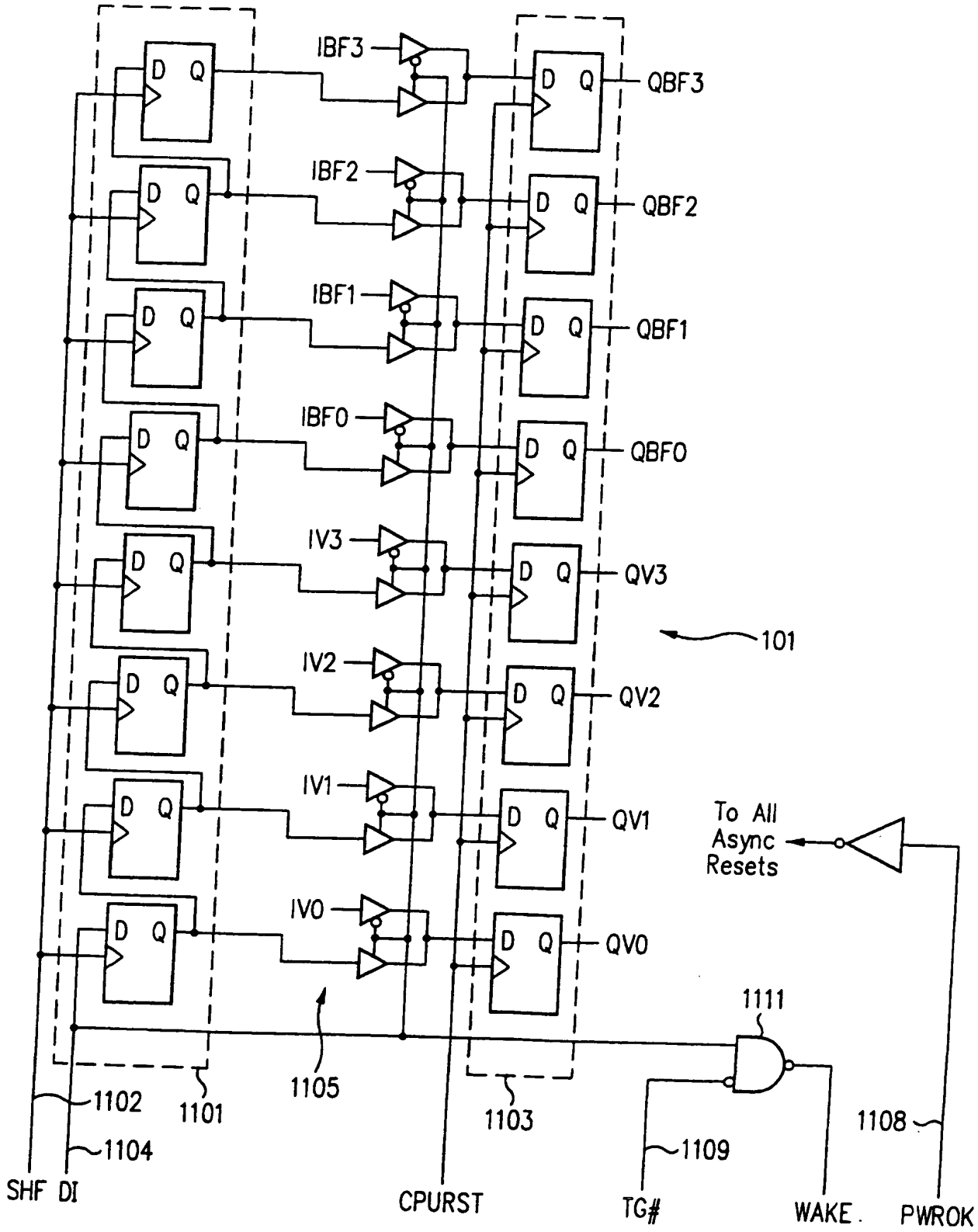

IBF3
IBF2
IBF1
IBF0
IV3
IV2
IV1
IV0
QBF3
QBF2
QBF1
QBF0
QV3
QV2
QV1
QV0
D Q
101
To All Async Resets
1111
1102
1101
1105
1103
1104
1109
1108
SHF DI
CPURST
TG#
WAKE.
PWROK

FIG. 11

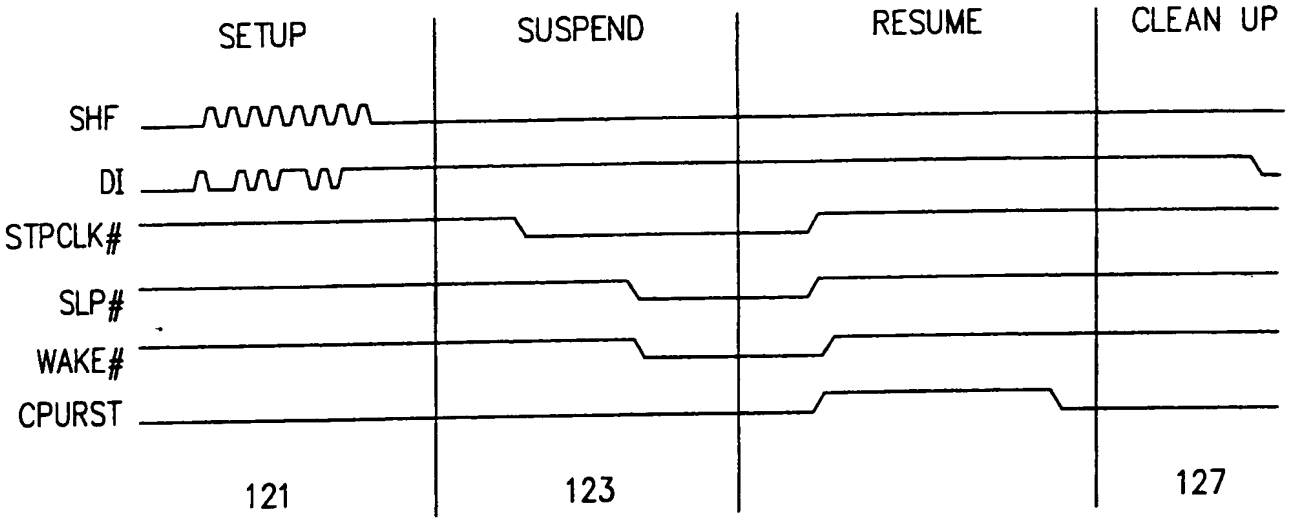


FIG. 12

130

(2) OPERATING MODE 131	(5) CORE VOLTAGE 132	(4) CPU CORE CLOCK MULTIPLIER CONFIGURATION 133	(1) RESET CONTROL BIT 134	(1) LATCH CMD BIT 135	(1) CLK STOP BIT 136

OPERATING MODE:
00=HIGH PERFORMANCE
01=AC POWER
10=BATTERY PERFORMANCE
11=BATTERY SAVE

FIG. 13

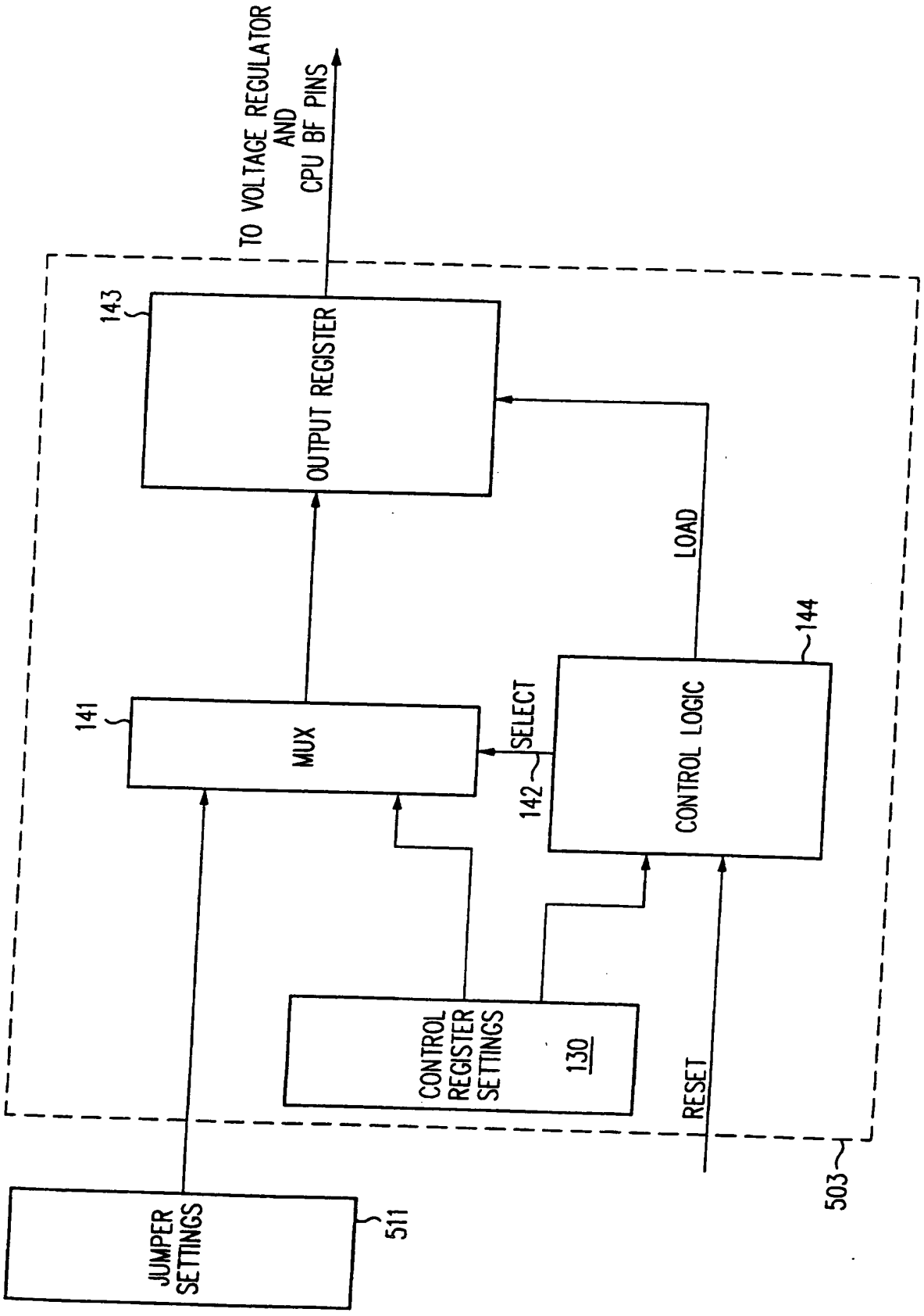

TO VOLTAGE REGULATOR AND CPU BF PINS
143
OUTPUT REGISTER
LOAD
144
CONTROL LOGIC
141
MUX
SELECT
142
CONTROL REGISTER SETTINGS
130
RESET
503
JUMPER SETTINGS
511

FIG. 14

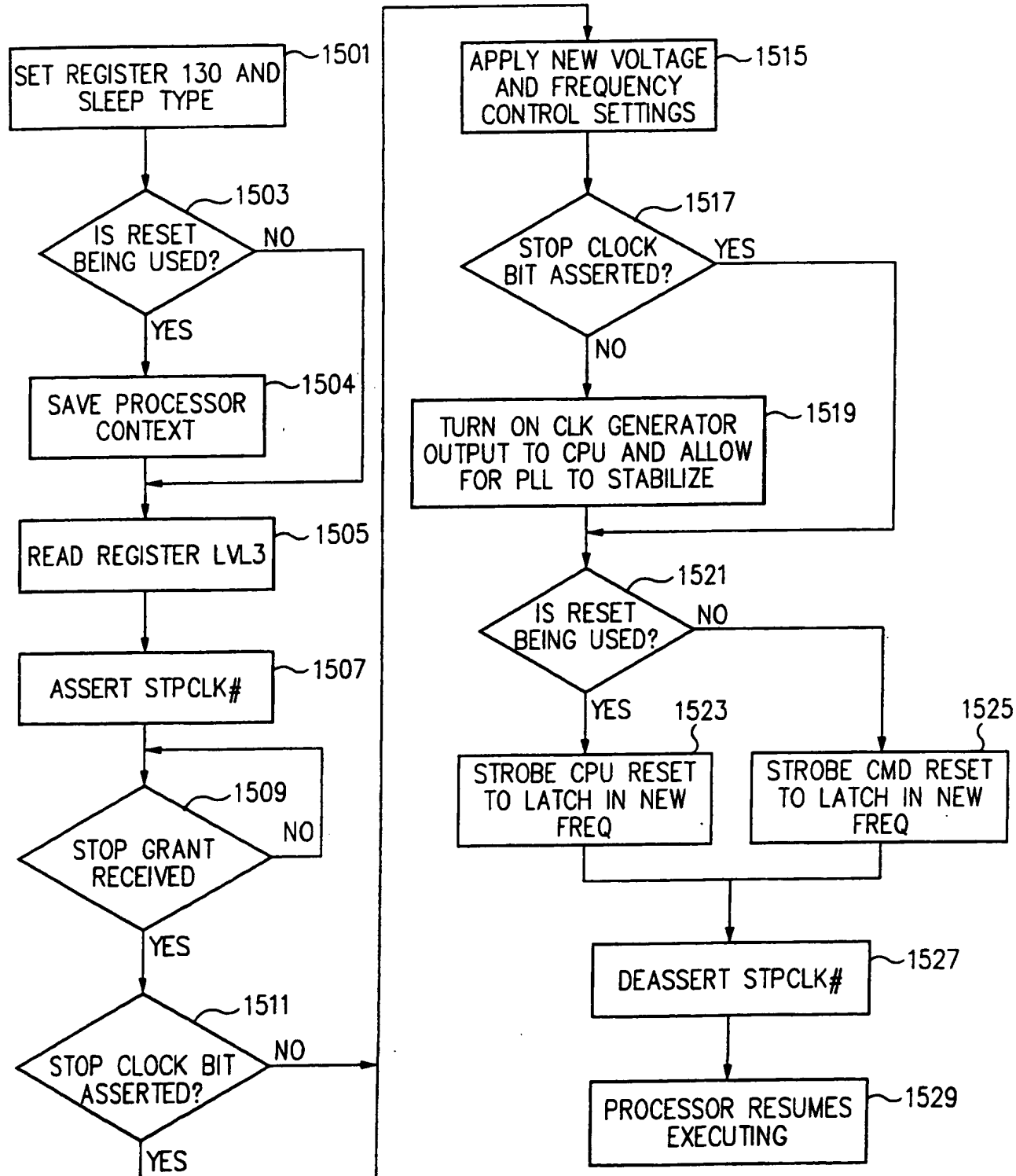

SET REGISTER 130 AND SLEEP TYPE
1501
IS RESET BEING USED?
1503
NO
YES
SAVE PROCESSOR CONTEXT
1504
READ REGISTER LVL3
1505
ASSERT STPCLK#
1507
STOP GRANT RECEIVED
1509
NO
YES
STOP CLOCK BIT ASSERTED?
1511
NO
YES
TURN OFF CLOCKS AT CLOCK SYNTHESIZER
1513
APPLY NEW VOLTAGE AND FREQUENCY CONTROL SETTINGS
1515
STOP CLOCK BIT ASSERTED?
1517
YES
NO
TURN ON CLK GENERATOR OUTPUT TO CPU AND ALLOW FOR PLL TO STABILIZE
1519
IS RESET BEING USED?
1521
NO
YES
1523
1525
STROBE CPU RESET TO LATCH IN NEW FREQ
STROBE CMD RESET TO LATCH IN NEW FREQ
DEASSERT STPCLK#
1527
PROCESSOR RESUMES EXECUTING
1529

FIG. 15

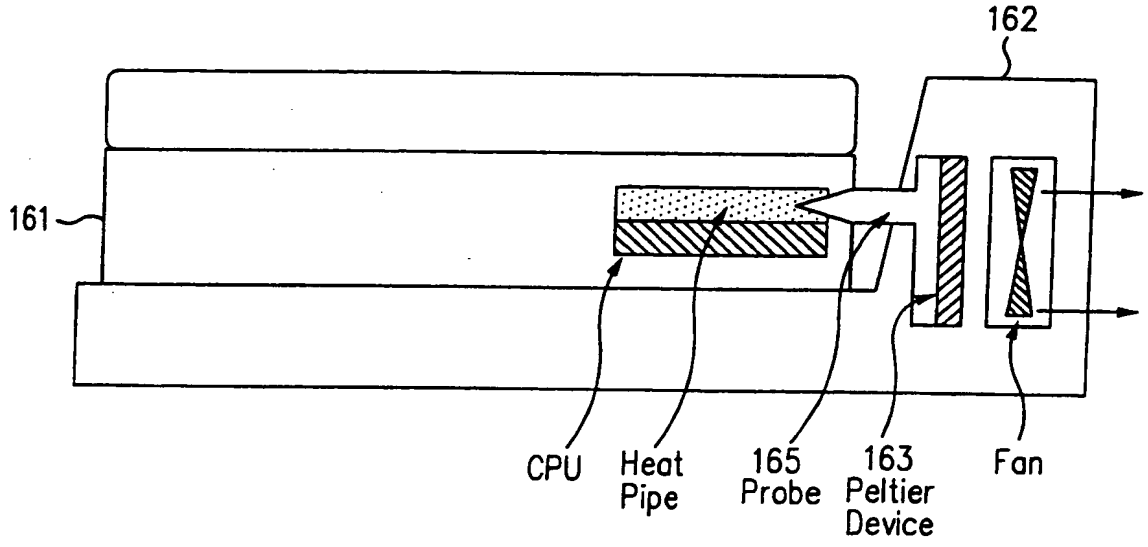


FIG. 16

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/17460

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/32 G06F1/20

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
WPI Data, EPO-Internal, IBM-TDB, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 887 179 A (ZMORA EITAN ET AL) 23 March 1999 (1999-03-23) column 1, line 46 -column 2, line 31 column 3, line 62 -column 4, line 48	10
Y	---	1-9
X	WO 97 12329 A (GUNTHER STEPHEN H ;SENYK BORYS (US); BHAT KETAN (US); INTEL CORP () 3 April 1997 (1997-04-03) page 3, last paragraph -page 9, paragraph 2ND	10
Y	---	11,12
	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 November 2000	12 01 2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Bailas, A

INTERNATIONAL SEARCH REPORT

Intc onal Application No
PCT/US 00/17460

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
L,P, Y	US 6 014 611 A (ARAI MAKOTO ET AL) 11 January 2000 (2000-01-11) FAMILY MEMBER IN NON-OFFICIAL LANGUAGE PUBLISHED ON TIME column 1, line 40 - line 65 column 3, line 25 -column 7, line 36 column 8, line 2 - line 10 column 12, line 35 - line 62	1-9,11, 12
Y	-& PATENT ABSTRACTS OF JAPAN vol. 1997, no. 4, 30 April 1997 (1997-04-30) & JP 08 328698 A (TOSHIBA CORP), 13 December 1996 (1996-12-13) abstract	1-9,11, 12
A	"POWER MANAGEMENT CLOCK CHANGE FOR 603 PROECESSOR" IBM TECHNICAL DISCLOSURE BULLETIN,US,IBM CORP. NEW YORK, vol. 38, no. 12, 1 December 1995 (1995-12-01), pages 325-327, XP000588158 ISSN: 0018-8689 the whole document	1,9
A	US 5 881 298 A (CATHEY DAVID A) 9 March 1999 (1999-03-09) page 1, line 50 -page 2, line 58 column 3, line 57 -column 5, line 52	6,8,11
A	US 5 925 133 A (BAUM GARY ET AL) 20 July 1999 (1999-07-20) column 4, line 49 -column 5, line 27	4

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter ›nal Application No
PCT/US 00/17460

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5887179 A	23-03-1999	NONE	
WO 9712329 A	03-04-1997	US 5745375 A	28-04-1998
		AU 7247296 A	17-04-1997
		EP 0858634 A	19-08-1998
		US 5825674 A	20-10-1998
US 6014611 A	11-01-2000	JP 3075957 B	14-08-2000
		JP 8328698 A	13-12-1996
US 5881298 A	09-03-1999	NONE	
US 5925133 A	20-07-1999	EP 0708406 A	24-04-1996
		JP 8263466 A	11-10-1996

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☑ ~~IMAGE~~ Text CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☑ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.